Reporting et archivage de données d'un SIGL

Erwann Mahé

▶ **To cite this version:**

Erwann Mahé. Reporting et archivage de données d'un SIGL. Cryptographie et sécurité [cs.CR]. 2012. dumas-01270865

HAL Id: dumas-01270865

https://dumas.ccsd.cnrs.fr/dumas-01270865

Submitted on 8 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

le cnam

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

CENTRE REGIONAL ASSOCIE DE STRASBOURG

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : INFORMATIQUE

OPTION : Informatique Système d'information

par

Erwann MAHE

Reporting et archivage de données d'un SIGL

Soutenu le 24 octobre 2012

JURY

PRESIDENT :	**Mme I.WATTIAU**	**Directrice département informatique**
MEMBRES :	**Mr C.KLEINPETER**	**Responsable pédagogique**
	Mr L.NEVEU	**Directeur service MI&CS, Lilly**
	Mr C.PFAFF	**Ingénieur CNAM**
	Mr R.REIBEL	**Ingénieur CNAM**

Remerciements

J'ai pu mener à bien le projet BOLDER R3.1 grâce la collaboration de nombreuses personnes. Tout d'abord, grâce la formation EuroLIMS et BOLDER qui m'a été dispensée par Alain Schneider, architecte technique de l'équipe Global MI&CS Lab.

C'est grâce à lui que j'ai pu monter en compétences sur ces deux applications et sur leurs composants techniques, pour finalement prendre totalement en charge seul la coordination du support technique d'EuroLIMS et de BOLDER.

L'expérience que j'ai pu acquérir dans ce cadre m'a permis de pouvoir diriger la réalisation du projet BOLDER R3.1.

Je remercie ensuite Ludovic Neveu, dont les interventions m'ont aidé à définir avec précision les besoins futurs des utilisateurs, mais aussi la stratégie d'avancement du projet et les évolutions futures de BOLDER. Je le remercie également pour ses relectures de mon mémoire et pour les conseils prodigués.

Le projet n'aurait pu aboutir sans l'aide de toute une équipe, je tiens donc à citer ici la collaboration de Marie Reiminger, Virginie Fix, Marie-Agnès Muñoz, Mehmet Akkus, Lilian Lhenry et Philippe Gramfort, ainsi que les équipes offshore[1] de Tata Consulting et d'IBM.

Enfin, je tiens à remercier monsieur Levra pour le soutien que m'a apporté Prodaxis, en terme de formation Business Objects, et monsieur Kleinpeter pour ses relectures et ses conseils concernant l'écriture de mon mémoire.

[1] Offshore : voir glossaire.

Glossaire

- Ada!Rec : application ETL développée par Lilly, utilisée pendant de nombreuses années, avant d'être progressivement remplacée par Informatica (ETL : voir liste des abbréviations).
- BOLDER : Business Objects for LIMS Data Extraction and Reporting. Application dédiée au reporting des données d'EuroLIMS.
- Crontab : commande Unix ou Linux permettant de lancer des commandes selon des horaires déterminés.
- Custodian (System) : dans l'environnement Lilly, personne responsable envers le System Owner du bon développement, de l'installation, de la maintenance et du support des systèmes, selon les règles et procédures de l'entreprise. Dans le cas d'une inspection, le Custodian a la responsabilité de présenter la documentation de validation spécifique à l'application. Il doit avoir l'autorité de prendre des décisions concernant la validation du système. On retrouve un System Custodian pour chaque application sur chaque site Lilly et également un Global Custodian dans le cas où cette application est utilisée sur plusieurs sites. Source : réglementation interne Lilly ('Roles and Responsibilities').
- IBM : International Business Machines. Société de services d'ingénierie informatique internationale, impliquée dans le projet BOLDER R3.1.
- Offshore : dans le contexte de l'informatique, l'offshore désigne le fait de confier à une équipe étrangère, située dans un autre pays, la réalisation d'une prestation informatique. Ceci dans le but de réduire les coûts de réalisation.
- Owner (Application ou System) : dans l'environnement Lilly, employé qui a la responsabilité de la validation du système tout au long de son cycle de vie, et qui a autorité sur les processus métiers et sur les données que le système supporte ou met en œuvre. Il est également responsable de tout résultat d'audit ou d'inspection sur le système dont il est en charge. On retrouve un System Owner pour chaque application sur chaque site Lilly et également un Global Owner dans le cas où cette application est utilisée sur plusieurs sites. Source : réglementation interne Lilly ('Roles and Responsibilities').
- Power User : dans l'environnement Lilly, utilisateur expert d'une application, qui utilise ses fonctionnalités les plus avancées et sert de référent aux autres utilisateurs.

- Reporting : compte-rendu. Dans le cadre de l'informatique, ce terme fait référence aux rapports créés à partir des données de production répliquées et dénormalisées, à l'aide d'une application de reporting telle que Business Objects ou Crystal Report.
- Snapshots : clichés. Dans le cadre d'une base de données relationnelle, instantané d'une série de données à un instant précis.
- TCS : Tata ConSulting. Société de services d'ingénierie informatique indienne, impliquée dans le projet BOLDER R3.1.
- Technical Leader : définit dans le cadre de ce mémoire par 'architecte technique', de manière à correspondre au plus prêt à la réalité des fonctions exercées.
- Triggers : déclencheurs. Dans le cadre d'une base de données relationnelle, action sur des données qui va induire une ou plusieurs autres actions sur d'autres données.
- Views : vues. Dans le cadre d'une base de données relationnelle, représentation d'une sélection de données en provenance d'où ou plusieurs autres tables.

Liste des abréviations

- BLOB (Binary Large Object) : type de champ de base de données relationnelle permettant de stocker des données binaires.
- CR - Change Request : dans le contexte Lilly, appellation donnée aux demandes de changements informatiques dans une application.
- ERP : Enterprise Resource Planning ou progiciel de gestion intégré. Système informatique de planification des ressources de l'entreprise.
- ETL (application) : Extract Transform and Load (Extraire, Transformer et Charger). Application informatique servant d'outil d'interface de données entre deux ou plusieurs applications distinctes.
- FIT : Formal Integration Testing : phase de tests formelle.
- IIT : Informal Integration Testing : phase de tests informelle.
- LIMS : Laboratory Information Management System (voir SIGL).
- MI&CS : Manufacturing Informatic and Computer Science (appelé également Global MI&CS), département informatique dont dépend l'équipe dont je fais partie, en charge du support de nombreuses applications informatiques.
- SIGL : Système d'Information de Gestion de Laboratoire. Progiciel dédié à la gestion et aux enregistrements des travaux de laboratoire. système informatique de gestion de laboratoire. Traduit en anglais par LIMS.
- SIPPQ : Sécurité, Intégrité, Principe, Pureté et Qualité. Ce sont les cinq caractéristiques d'un produit pharmaceutique, conforme aux Bonnes Pratiques de Fabrication. Si l'un des ces caractéristiques est dégradée, le produit est déclaré non-conforme.
- SISPQ : Safety, Integrity, Strength, Purity and Quality (voir SIPPQ).
- SSII : société de services en ingénierie informatique.
- TPR - Test Problem Report. Il s'agit de l'investigation et de la résolution d'un problème rencontré lors d'un test exécuté.
- VGL : Vacuum Gauge Language. langage de programmation propriétaire de Thermo Scientific, utilisé dans EuroLIMS (Sample Manager).

Table des matières

Remerciements 2
Glossaire 3
Liste des abréviations 5
Table des matières 6
Introduction 9
I PREMIERE PARTIE : LE CONTEXTE DU PROJET 10
I.1 CHAPITRE UN : L'ENVIRONNEMENT PROFESSIONNEL 10
I.1.1 Le groupe Prodaxis 10
I.1.2 Parcours professionnel 11
I.1.3 Eli Lilly and Company 12
I.1.3.1 Une entreprise centenaire 12
I.1.3.2 Une compagnie pharmaceutique internationale 13
I.1.4 L'équipe Global MI&CS Lab 14
I.1.5 Bangalore (Inde) : 15
I.1.6 Fegersheim (Alsace) : 15
I.2 CHAPITRE DEUX : LE CONTEXTE INFORMATIQUE 17
I.2.1 Le processus des métiers de laboratoire 17
I.2.2 Qu'est-ce qu'un LIMS/SIGL ? 19
I.2.3 Le SIGL EuroLIMS 19
I.2.3.1 Une application évolutive 20
I.2.3.2 Champ d'application du SIGL 20
I.2.3.3 Une application critique 25
I.2.3.4 Volumes d'utilisation d'EuroLIMS 25
I.2.4 L'outil de reporting BOLDER 26
I.2.4.1 L'architecture BOLDER 27
I.2.4.2 l'outil d'interface 27
I.2.4.3 La base de données 29
I.2.4.4 Business Objects 30
I.2.5 La fin de vie d'EuroLIMS et son futur retrait 30
II DEUXIEME PARTIE : LE PROJET BOLDER R3.1 32
II.1 CHAPITRE UN : DEFINITION DU PROJET 32
II.1.1 Enoncé 32
II.1.2 Objectifs du projet 33
II.1.2.1 Objectifs techniques et fonctionnels 33
II.1.2.2 Objectifs de validation 34
II.1.2.3 Objectif documentaire 34
II.1.3 Les contraintes externes au projet 35
II.1.4 Les contraintes internes au projet 36
II.1.4.1 Contrainte de ressources limitées 36
II.1.4.2 Contrainte de qualité 38
II.1.4.3 Contrainte de temps 38
II.2 CHAPITRE DEUX : LA MISE EN PLACE DU PROJET 40
II.2.1 Le recueil des besoins 40
II.2.2 La méthode de projet 42

II.2.2.1 Etat des lieux 42
II.2.2.2 La méthode de projet du Global MI&CS Lab 45
II.2.3 Le périmètre des responsabilités du chef de projet 53
II.2.4 L'équipe des débuts 55
II.2.5 Un démarrage assez lent 55
II.2.6 L'équipe renforcée 56
II.2.7 La gestion des priorités de la mission 57
II.2.7.1 Temps partagé entre le support et le mode projet 57
II.2.7.2 Favoriser l'autonomie des analystes 58
II.3 CHAPITRE TROIS : L'ECRITURE DU PLAN DE PROJET 59
II.3.1 Définition des grandes étapes 59
II.3.2 Importance relative de la date de livraison 61
II.3.3 Difficiles estimations de temps 62
II.3.4 Renouvellement de la contrainte temps 63
II.4 CHAPITRE QUATRE : LES PREMIERES ETAPES TECHNIQUES 65
II.4.1 Les tests de compatibilité Ada!Rec vs Oracle 11 65
II.4.1.1 Le contexte 65
II.4.1.2 Les tests effectués 65
II.4.2 Migration de la plate-forme Business Objects 66
II.4.3 La formation à Business Objects XI r3 66
II.4.4 L'écriture du Release Impact Assessment 67
II.4.5 La modélisation de la base de données 68
III TROISIEME PARTIE : MIGRATION ET ECRITURE DES TESTS 70
III.1 CHAPITRE UN : MIGRATION DES TESTS VERS UNE SOLUTION ELECTRONIQUE 70
III.1.1 Situation préexistante 70
III.1.2 Les raisons de la migration 70
III.1.3 La migration des cas de tests BOLDER 71
III.2 CHAPITRE DEUX : (RE)ECRITURE DES NOUVEAUX CAS DE TESTS 72
III.2.1 Refonte du système d'identification des tests 72
III.2.2 Réécriture des tests 72
III.2.3 Mise à jour de la documentation relative aux tests 74
IV QUATRIEME PARTIE : DEVELOPPEMENT DE L'APPLICATION BOLDER 77
IV.1 CHAPITRE UN : ORGANISATION DES ACTIVITES 77
IV.1.1 La problématique du suivi des fichiers 77
IV.1.1.1 Besoin d'une solution de suivi simple et commune 78
IV.1.1.2 Le processus au sein de l'équipe Global MI&CS Lab 79
IV.1.1.3 La solution SharePoint 79
IV.1.1.4 Les logiciels du marché 81
IV.1.1.5 Mise en place d'un nouveau processus du suivi de fichiers 81
IV.1.2 Méthodes de travail 83
IV.1.2.1 Importance du respect du modèle de la base de données 83
IV.1.2.2 Systématisation des tests unitaires 84
IV.1.2.3 Nécessité d'analyse en profondeur 85
IV.1.3 Cas de recherche et développement 88
IV.2 CHAPITRE DEUX : DEVELOPPEMENTS TECHNIQUES 92
IV.2.1 L'écriture des spécifications techniques 92
IV.2.1.1 Confirmation directe des modèles 92
IV.2.1.2 Une revue plus simple 93
IV.2.2 Le développement de l'interface et de la base de données 94
IV.2.3 Les artefacts Business Objects 95

IV.2.3.1 un travail plus conséquent que prévu 95
IV.2.3.2 une expertise technique et métier nécessaire 96
V CINQUIEME PARTIE : PREPARATION DES TESTS ET ETAPES A VENIR 97
V.1 CHAPITRE UN : ACCELERATION DU PROJET 97
V.1.1 Renfort de ressources 98
V.1.2 Avancer l'exécution des tests 99
V.1.3 Le nouvel ordonnancement des tests 100
V.1.4 Finalisation de la documentation de testing 102
V.1.5 Préparation de l'environnement 104
V.2 CHAPITRE TROIS : LA SUITE DU PROJET 104
V.2.1 Prochaines étapes 104
V.2.2 Préparer la future version de BOLDER R3.2 105
Conclusion 107
Table des annexes 113
Annexe 1 - Extrait n°1 du cahier des charges BOLDER R3.1 114
Annexe 2 - Extrait n°2 du cahier des charges BOLDER R3.1 115
Annexe 3 - Méthode de gestion de projet en vigueur au sein du Global MI&CS Lab 116
Annexe 4 - Déroulé temporel des actions du projet 117
Annexe 5 - Extrait du plan de projet 118
Annexe 6 - Extrait du document de design de la base de données : table TEST 119
Annexe 7 - Extrait des normes de codage pour le langage Ada!Rec 120
Annexe 8 - Extrait du script Ada!Rec 'adapter' pour la table des marchés. 121

Introduction

Cadre technicien informatique au sein de la SSII Prodaxis Grand-Est, je travaille en tant que consultant auprès de la société pharmaceutique Eli Lilly.

Je suis chargé de la coordination du support de deux applications informatiques utilisées par les équipes laborantines Lilly en Europe : d'une part le système informatique de gestion de laboratoire (SIGL) EuroLIMS, d'autre part la solution de reporting qui y est associée, BOLDER[2].

Dans le cadre de ma mission, j'ai été amené à diriger et coordonner le projet d'évolution de BOLDER, en parallèle de mes autres activités.

La raison de cette évolution est double : d'une part, valider les nouveaux composants techniques nécessaires au fonctionnement de cette application, d'autre part, intégrer un nombre plus important de données venant du SIGL[3] EuroLIMS.

Il est en effet prévu de retirer et clore celui-ci : cette action consistera dans un premier temps à stopper toute écriture dans la base de données, à couper tout accès et à mettre à jour la documentation de validation en rapport.

La solution de reporting BOLDER deviendra alors une solution de stockage et de présentation de données archivées, ceci pour plusieurs années après la fin de l'utilisation d'EuroLIMS. Ce projet a débuté en février 2011, avec une équipe réduite, et est encore en cours actuellement. La rédaction de ce mémoire s'est arrêtée à l'état du projet de mi-août 2012. Le lecteur trouvera en annexe 4 un déroulé temporel des principales actions du projet.

Nous verrons donc dans ce mémoire le contexte dans lequel il se déroule, les champs d'application concernés, et sa mise en place. Puis nous verrons le rôle que j'ai assuré dans les phases de création et développement, et de préparation des tests. J'aborderai enfin les futures étapes des tests, de livraison, de déploiements en production et de formations aux utilisateurs, avant de conclure.

En raison du contexte dans lequel je travaille, un grand nombre de termes techniques et professionnels sont utilisés en anglais. J'ai pris le soin dans ce mémoire de les restituer en français, si cela ne dénuait pas leur sens et si une traduction existait.

[2] BOLDER : Business Objects for LIMS Data Extraction and Reporting. Application dédiée au reporting des données d'EuroLIMS.
[3] Système informatique de gestion de laboratoire. Voir la liste des abréviations.

I Première partie : le contexte du projet

I.1 Chapitre un : l'environnement professionnel

Avant de pouvoir décrire le déroulement du projet BOLDER R3.1, il convient d'introduire l'environnement professionnel dans lequel il s'est réalisé.

Nous verrons ainsi une présentation de la société Prodaxis pour laquelle je travaille en tant que consultant informatique. Puis j'évoquerai mon parcours professionnel, qui m'a conduit à assurer des prestations informatiques chez Eli Lilly, société que je présenterai ensuite.

J'expliquerai enfin plus précisément le rôle des applications EuroLIMS et BOLDER maintenues et supportées par mon équipe. Nous verrons dans ce cadre comment la fin de vie et le futur retrait d'EuroLIMS a amené le besoin de faire évoluer l'application de reporting BOLDER.

I.1.1 Le groupe Prodaxis

Fondée en 1990, la société Prodaxis, basée à Lyon, est à l'origine l'éditeur de l'ERP[4] Parteor, dédié aux sites industriels et entreprises de taille moyenne, dont la première version a été mise sur le marché en 2003. En 2006, la version 1.5 est commercialisée, la version 2.0 le sera en 2010, la version 2.5 en 2012.

En 2007 l'entreprise Prodaxis est devenue le groupe Prodaxis, en rachetant la société Orgalis dont je faisais alors partie, société basée à Strasbourg. Cette dernière était plus spécialisée dans l'offre de services informatiques (support et maintenance matériels, régie), la vente de progiciels de gestion et le développement de sites internet.

Le Groupe Prodaxis compte aujourd'hui une trentaine de collaborateurs répartis dans les agences lyonnaise et strasbourgeoise. Ses activités se concentrent notamment sur l'informatique de gestion et les solutions internet (e-business, gestion de contenu), qui représentent deux tiers de son activité.

[4] ERP : Enterprise Resource Planning ou progiciel de gestion intégré. Voir la liste des abréviations.

Parmi les clients, nous retrouvons de grands noms de l'industrie : Bosch, EADS, Lilly, Rolls Royce, ST-Ericsson, Thales et des institutions telles que l'Assemblée nationale, l'Union des Eglises Protestantes d'Alsace Lorraine, le Théâtre national de Strasbourg.

Afin de proposer des services informatiques sur des secteurs informatiques bien définis, l'offre de Prodaxis se répartit de la manière suivante :

Tableau 1 : répartition des offres de service de Prodaxis

Infrastructure	Consulting	Progiciels de gestion	Logiciels de reporting	E-Business
Vente ou location Equipement Maintenance	Mise en œuvre Développement Test/homologation Formation	Edition Vente Support Formation	Développement Intégration	Développement de solutions de commerce électronique

Je fais partie de la branche 'Consulting' et suis rattaché à l'entité Prodaxis Grand-Est, dont le champ d'activité est notamment l'Alsace. Je suis à ce titre actuellement en mission de support et projet informatique chez Eli Lilly, à Fegersheim.

I.1.2 Parcours professionnel

J'ai commencé ma carrière d'informaticien en tant que développeur web en 2003, métier que j'ai effectué pendant trois ans.

Puis j'ai intégré en 2006, au sein de la société Orgalis une équipe de support niveau un, en prestation chez Lilly France, sur le site de Fegersheim (Alsace), pour deux missions consécutives.

Au bout d'un an et demi, je suis passé dans le domaine de la validation informatique, toujours chez Lilly France, sur des applications utilisées dans les zones de production et de conditionnement de médicaments.

Par la suite, j'ai été amené à prendre en charge le support de niveau deux d'une application de maintenance, au sein de l'équipe européenne de support informatique Global MI&CS[5] .

[5]MI&CS: Manufacturing Information & Computer Science.

Mon travail a consisté pendant huit mois à coordonner l'activité d'une équipe d'analystes offshore située à New Dehli en Inde, à communiquer régulièrement avec les sites européens utilisateurs afin de mieux comprendre leurs besoins et à remonter ceux-ci à la partie de l'équipe basée à Indianapolis, Etats-Unis d'Amérique, celle-ci décidant des orientations stratégiques, techniques et des processus d'utilisation du système.

Cette mission m'a permis d'acquérir une expérience du support informatique dans un contexte international, et j'ai pu ainsi intégrer ensuite l'équipe Global MI&CS Lab, où j'effectue depuis Janvier 2009 ma mission actuelle.

I.1.3 Eli Lilly and Company

I.1.3.1 Une entreprise centenaire

Eli Lilly est une compagnie pharmaceutique américaine à dimension internationale, fondée à Indianapolis en 1876 par le colonel Lilly, vétéran de l'armée fédéral des Etats-Unis d'Amérique[6].

Les médicaments de l'époque étaient très souvent des produits mal préparés et inefficaces, voire nocifs. Le colonel Lilly créé alors une société pharmaceutique dont le but est de délivrer des médicaments de la meilleure qualité, uniquement sur le conseil de médecins.

La société s'est ensuite développée et le colonel Lilly décide d'améliorer d'une part les méthodes de fabrication des médicaments, pour assurer leur qualité, et d'autre part de miser sur la recherche de médicaments plus efficaces.

Ce sont ces deux valeurs qui contribuent aujourd'hui à la culture de l'entreprise :

- intégrité
- excellence
- s'y ajoute la troisième valeur de Lilly : la volonté de respecter les personnes ; que ce soit les clients, qui accordent leur confiance à l'entreprise, pour pouvoir se soigner, ou les collaborateurs qui contribuent au bon fonctionnement de l'entreprise et à l'atteinte de ses objectifs.

[6] http://www.lilly.fr/lilly/histoire.cfm

Deux innovations pharmaceutiques majeures, parmi beaucoup d'autres, ont été faites par la compagnie Eli Lilly : la première insuline industrielle en 1923 et la première pénicilline industrielle en 1941.

I.1.3.2 Une compagnie pharmaceutique internationale

Lilly développe, produit et vend aujourd'hui des médicaments destinés notamment au traitement des maladies psychiques (dépression, schizophrénie), du cancer, du diabète. Un autre pan de son activité concerne la production de médicaments à usage vétérinaire.

De par ses activités de recherche, de production et de vente, Lilly est classée au neuvième rang mondial des laboratoires pharmaceutiques et cinquième compagnie en biotechnologie.

Cinq de ses treize sites de production se trouvent en Europe, dont trois dédiés aux produits finis (France, Espagne, Italie).

Le site de Fegersheim (Alsace), où j'effectue ma mission, est le premier site de production Lilly en termes de taille[7], pour la fabrication de produits injectables.

Un cinquième de son chiffre d'affaires est consacré à la recherche et au développement de nouveaux médicaments et de protéines biologiques, dans huit centres aux Etats-Unis, au Japon et en Europe. Lilly est implantée dans soixante-treize pays et vend ses médicaments dans cent vingt-cinq états[8].

[7] http://www.lilly.fr/lilly/industrie-pharmaceutique.cfm
[8] http://www.lilly.fr/lilly/groupe-pharmaceutique.cfm

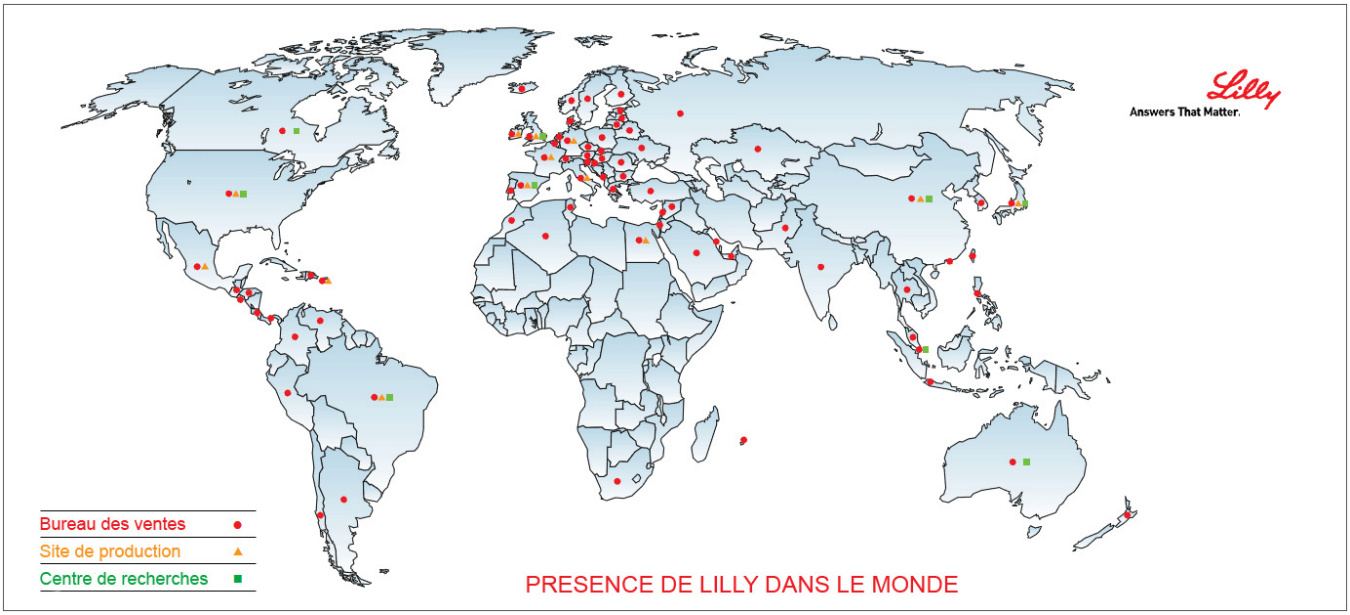


Figure 1- Carte des implantations des établissements Lilly dans le monde

I.1.4 L'équipe Global MI&CS Lab

Ce département informatique regroupe les équipes de support d'une demi-douzaine d'applications de laboratoires, utilisées en Europe, en Chine et sur le continent américain.

Ce qui permet à l'ensemble des membres de mettre en commun bon nombre de connaissances et de pouvoir collaborer et s'épauler en cas d'investigations communes.

Elle n'est pas rattachée au site de Fegersheim, mais dépend d'une équipe américaine, responsable d'applications utilisées par de multiples sites Lilly de par le monde. La stratégie de support, de maintenance et de cycle de vie n'est plus locale, mais globale et s'applique par zones géographiques.

Le Global MI&CS Lab pour la zone Europe est dirigée par le chef de service Lilly Ludovic Neveu, qui remplit entre autres les fonctions de « Global Custodian[9] » pour les applications EuroLIMS et BOLDER.

Ce rôle consiste à assurer le bon fonctionnement, l'évolution, le support la validation et la maintenance des deux systèmes, pour l'ensemble des sites européens concernés, ceci dans le respect des procédures Lilly.

[9] Global Custodian : voir glossaire.

Le Global Custodian s'appuie pour le support des deux applications EuroLIMS et BOLDER sur deux cadres Lilly et sur un contrat de service passé avec un prestataire. Ainsi une demi-douzaine de personnes compose l'équipe de support, basées en trois lieux distincts :

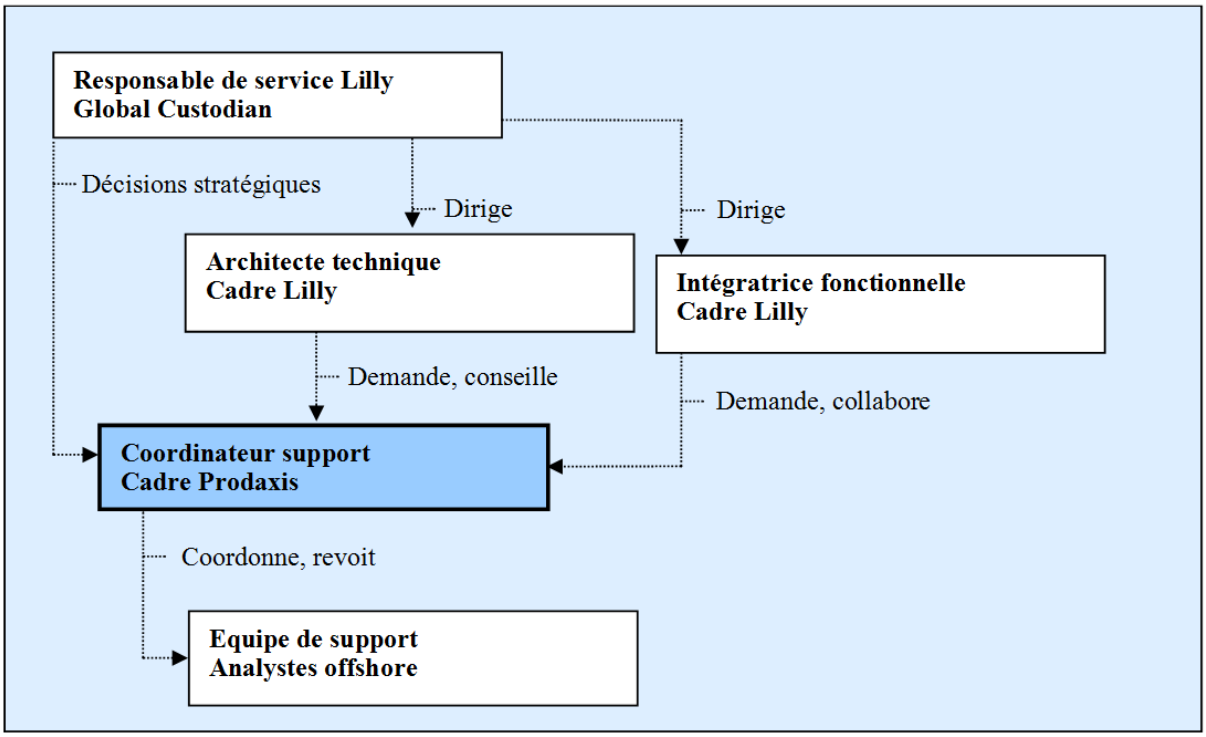


Figure 2 - Structure schématique de l'équipe support EuroLIMS/BOLDER

I.1.5 Bangalore (Inde) :

Quatre consultants informatiques assurent le support opérationnel, les développements, les installations et la mise à jour documentaire. Ils ont été formés à l'ensemble des procédures et des modes opératoires nécessaires au support d'EuroLIMS et BOLDER. Ces analystes ont endossé ces fonctions de support au mois de mai 2012, en remplacement du prestataire précédent.

I.1.6 Fegersheim (Alsace) :

Deux cadres Lilly (un architecte technique, une intégratrice fonctionnelle) interviennent des questions précises qui demandent des connaissances pointues, une forte connaissance des applications EuroLIMS/BOLDER ou de la pratique des métiers de laboratoires.

Je suis pour ma part chargé de la coordination de l'équipe offshore citée ci-dessus : j'organise, coordonne et revois les activités techniques de support et de maintenance et je suis également chargé de faire l'interface entre les sites utilisateurs et l'équipe de support offshore.

J'interviens en cas d'absence de l'équipe offshore, ou dans le cas où l'activité technique requiert une plus grande proximité avec les sites utilisateurs ou l'équipe Global MI&CS Lab.

Enfin, je remplis le rôle de point de contact auprès des équipes de support d'infrastructures informatiques.

I.2 Chapitre deux : le contexte informatique

Afin de pouvoir comprendre le fonctionnement du SIGL EuroLIMS, intéressons-nous aux processus métiers qui nécessitent son emploi, ainsi qu'une définition de ce qu'est qu'un SIGL. Puis nous verrons le fonctionnement de chacune des deux applications supportées. Enfin, nous verrons comment le futur retrait d'EuroLIMS a entraîné le besoin de faire évoluer BOLDER en outil de recherche de données archivées.

I.2.1 Le processus des métiers de laboratoire

L'activité principale de Lilly est de produire et de vendre des médicaments à un nombre important de pays de par le monde. Chaque pays ou groupe de pays (exemple : Union Européenne) constitue donc un marché, pour lequel sont définies un certain nombre de règles d'acceptation du médicament.

En cas de non-respect de ces règles, les médicaments ne peuvent obtenir l'autorisation de mise sur le marché. S'ajoutent à ces réglementations les règles que Lilly impose à ses propres produits ou à ceux de ses fournisseurs. Il est donc impératif pour cette société pharmaceutique de s'assurer, à toutes les étapes de production, de stockage et d'avant-expédition, de la qualité des produits et de la stabilité de celle-ci, en les testant.

Tableau 2 : répartition des activités de tests laborantins dans les processus métiers

Direction	Qualité	Production	Logistique/Stockage	Maintenance	Supports
/	Tests selon les standards Lilly ou les lois des pays marchés Tests sur produit acheté Tests environnementaux Décisions de la qualité d'un lot	Tests sur produit semi-fini Tests sur produit fini	Tests sur stabilité de produits stockés selon les conditions de stockage, dans le temps	/	/

Les processus des métiers de laboratoires dépendent bien évidemment de la direction Qualité. Cependant, certains se retrouvent associés ou intégrés à trois des processus métiers de Lilly.

Nous trouverons dans le schéma qui suit, un aperçu des principaux processus intervenants dans la fabrication d'un lot de médicaments, montrant ainsi les étapes où interviennent les tests en laboratoire.

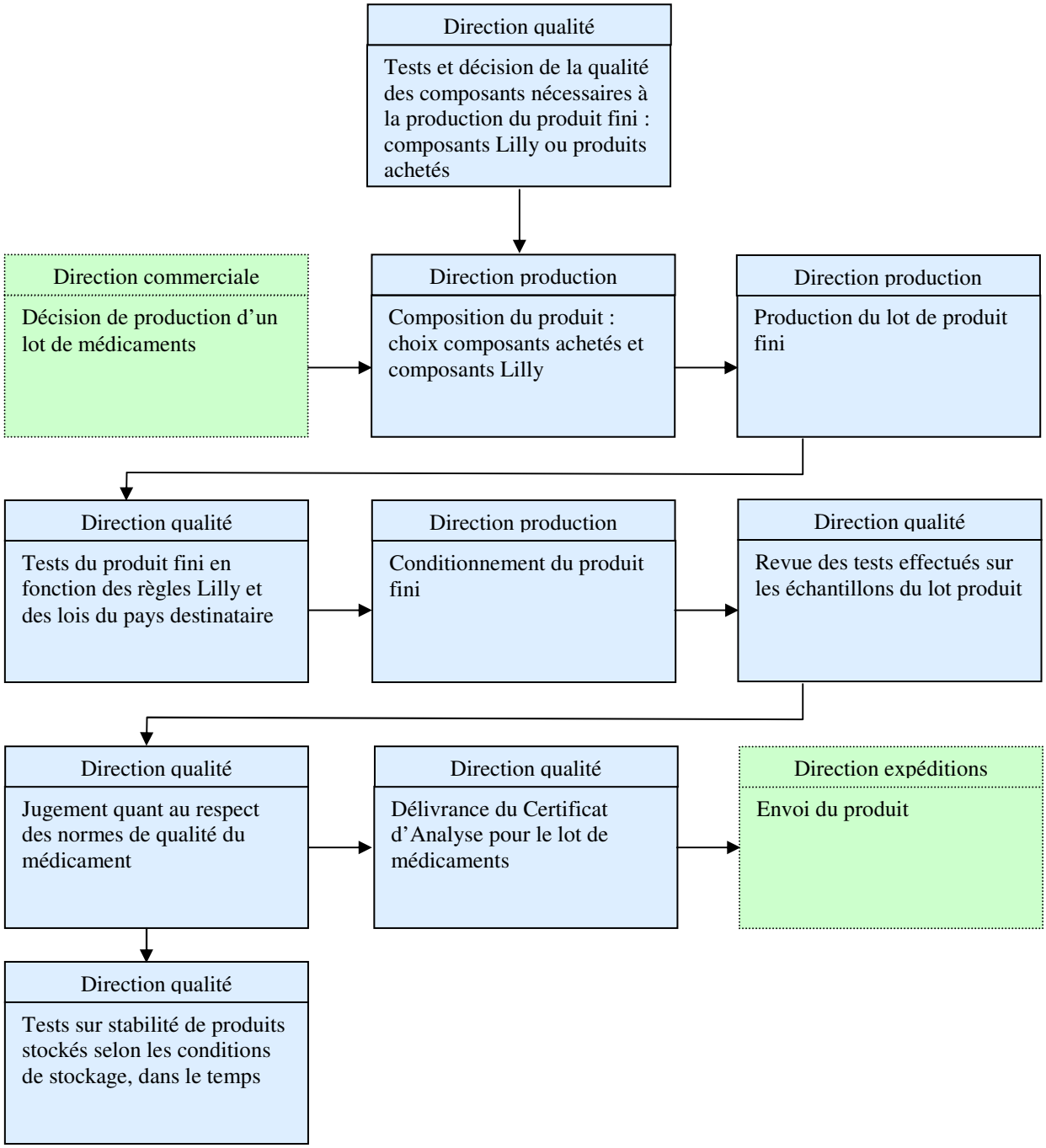


Figure 3 - Schéma des processus des métiers de laboratoires, intervenant lors de la fabrication d'un lot de produit

I.2.2 Qu'est-ce qu'un LIMS/SIGL ?

Les actions relevées plus haut dans les processus doivent pouvoir être effectuées par les équipes laborantines dans un système informatique de gestion adapté à leurs besoins. Ce que permettent les progiciels de type LIMS (acronyme de Laboratory Information Management System ou SIGL pour Système Informatique de Gestion de Laboratoire).

Un SIGL permet en premier lieu de tracer et de tester les échantillons de laboratoire et enregistrer les résultats de tests.

Il permet aussi de définir des stratégies de tests, de prévoir les séries de prélèvements, de tenir à jour les stocks, de confirmer la qualité des produits des fournisseurs. S'ajoutent les fonctions de base de tout progiciel : gestion des utilisateurs, des données de dimensions (emplacements, instruments, unités de mesure,..).

Le SIGL est un progiciel essentiel pour le bon fonctionnement des laboratoires, en raison de la masse de données croissante à traiter et conserver, mais aussi pour répondre aux demandes des organismes d'inspection. Il permet ainsi aux équipes laborantines de prouver aux agences sanitaires la qualité et la traçabilité des échantillons testés.

I.2.3 Le SIGL EuroLIMS

EuroLIMS est un système d'information de gestion de laboratoire développé chez Lilly, par l'équipe Global MI&CS Lab, à partir du progiciel Sample Manager vendu par l'éditeur Thermo Scientific.

Lilly a déployé au milieu des années 1990 la première version de Sample Manager, a implémenté au début des années 2000 la version R2.0 et ensuite a ajouté différentes fonctionnalités au fil du temps, selon les besoins des sites et des évolutions de l'architecture des systèmes d'informations de Lilly.

On compte au total sept instances de production EuroLIMS ayant utilisé la version R2.0, chacune déployée sur un site européen distinct.

I.2.3.1 Une application évolutive

EuroLIMS est un progiciel qui a évolué avec le temps, au sein de Lilly, grâce à la structure de Sample Manager qui a permis ces évolutions, et à son langage non compilé, le VGL[10]. Ce langage ne représente pas le cœur de l'application, développée en code compilé, mais il est prévu pour mettre en place une grande partie des fonctions dédiées aux utilisateurs.

Pouvant disposer du code source de ces fonctions et pouvant rajouter d'autres modules, l'équipe Global MI&CS Lab a pu ainsi faire évoluer EuroLIMS, jusqu'en 2009, et le faire communiquer avec six systèmes externes différents.

Ainsi a été par exemple rajouté un module de gestion des données envoyées vers SAP, ou venant de cet ERP.

Plus récemment, un autre module a été développé pour permettre à EuroLIMS de « déléguer » à une application externe l'entrée des résultats de tests.

Les données des échantillons à tester sont ainsi envoyées vers une application externe, avec des champs de résultats vides. Une fois ceux-ci renseignés par les utilisateurs ou par des équipements électroniques (par exemple, des machines de pesée), les données (échantillons, tests et résultats) sont renvoyées vers EuroLIMS.

I.2.3.2 Champ d'application du SIGL

EuroLIMS a pour fonction de permettre aux équipes laborantines de renseigner les résultats des analyses effectuées sur des échantillons de produits semi-finis (solution médicamenteuse ou composants matériels), de relevés environnementaux et de médicaments produits sur le site.

Afin de comprendre la complexité de l'imbrication des données, je vais décrire dans les lignes qui suivent la logique au cœur du SIGL. Ce qui permettra de visualiser une partie des difficultés techniques lors du développement de BOLDER R3.1.

[10] VGL : Vacuum Gauge Language. Langage de programmation propriétaire de Thermo Fischer, ouvert aux équipes de support et maintenance de l'application.

La base de la structure de données, d'un point de vue utilisateur, est donc le triptyque 'échantillon-test-résultat'.

①

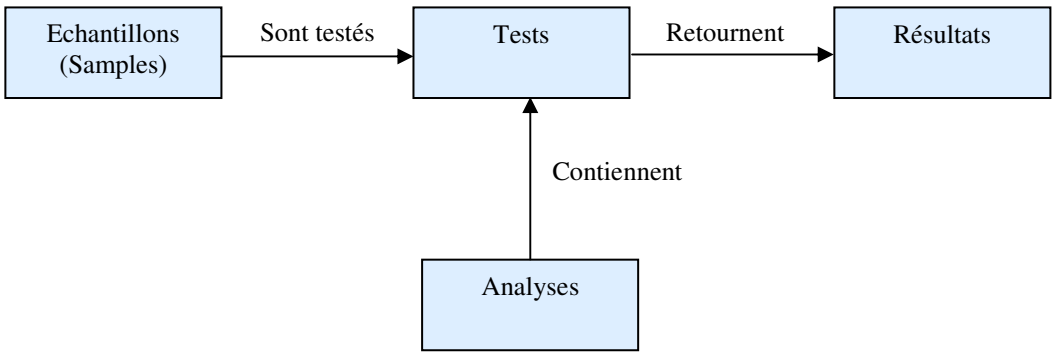


Figure 4 - Fonctionnement de base des tests dans EuroLIMS

Les tests qui doivent être effectuées sont définis au départ au travers d'une analyse et sont considérés comme ses composants.

C'est lors de la création de l'échantillon dans le système EuroLIMS que l'on va décider quelle(s) analyse(s) lui associer. Les composants de cette dernière vont donc servir à tester l'échantillon.

S'ajoute ensuite une structuration plus élaborée, en fonction de l'organisation des tests et des stratégies de tests, par produit testé, selon les exigences internes de Lilly et les exigences des marchés de destination (normes du pays).

Prenons un exemple pour mieux illustrer cette explication : un lot de médicament vient de sortir de la chaîne de production et est envoyé au magasin pour stockage. Une série d'échantillons est prélevé pour test.

Dans EuroLIMS, l'entrée des résultats pour l'échantillon se fera en rapport avec les tests effectués (température, PH, taux d'humidité, ou analyse plus spécifique). Ces résultats seront comparés aux limites fixées par Lilly et par la réglementation des pays marchés de destinations.

A noter que les limites fixées par Lilly sont très souvent plus drastiques que celles des pays destinataires.

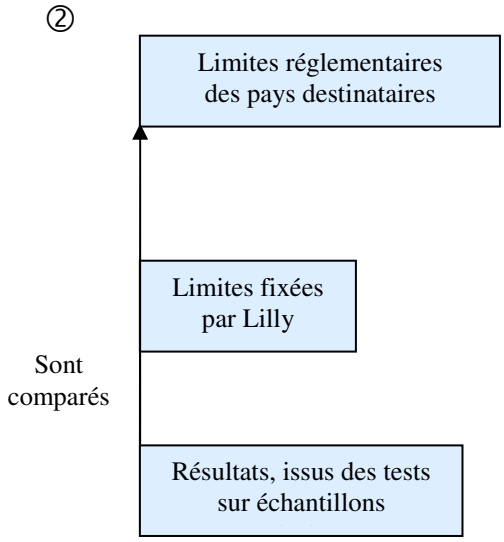


Figure 5 - Système de comparaison des résultats aux limites

Si un des résultats rentré n'est pas conforme à la limite correspondante (par exemple, PH trop élevé ou trop faible), le résultat sera déclaré hors-limites et cette information se retrouvera lors de la revue du lot de médicaments produits. Les équipes de la direction Qualité statueront alors dans EuroLIMS sur la disposition accordée au lot (approuvé ou rejeté), qui détermine s'il peut être mis en vente et expédié.

On le comprend aisément, la masse d'échantillons testés effectués, ainsi que la quantité de résultats entrés et revus, impliquent une structuration des données, qui permette aux utilisateurs de travailler efficacement.

Afin de pouvoir définir à l'avance et réutiliser le plus possible les données de limites, EuroLIMS donne aux utilisateurs la possibilité d'enregistrer les tests nécessaires et les limites auxquels leurs résultats seront comparés, dans une entité de données nommée 'Quality Master'.

Ces Quality Masters sont utilisés pour tester les lots de produits finis destinés à la vente, ou ceux de produits semi-finis. Ces matières sont soit des produits fabriqués par Lilly et testés avant d'être utilisés pour du produit fini, soit des produits livrés par des fournisseurs externes.

Ces entités de données fixent les limites de tests en fonction de trois critères :

- l'identifiant de l'article

- le pays source : il s'agit du lieu de fabrication du produit.

- la destination : il s'agit du pays ou de la zone pour laquelle le produit est destiné. Les tests effectués dépendent de cette destination. La réglementation des pays de destination impose en effet des tests précis sur le produit, afin qu'il soit accepté par les autorités sanitaires de ce pays.

Il est à noter qu'ils sont dotés d'un cycle de vie (numéro de version, statut de validité) comme beaucoup de données métiers dans EuroLIMS. Une fois un nouveau Quality Master approuvé, son prédécesseur est considéré comme obsolète et n'est plus utilisé.

②‘

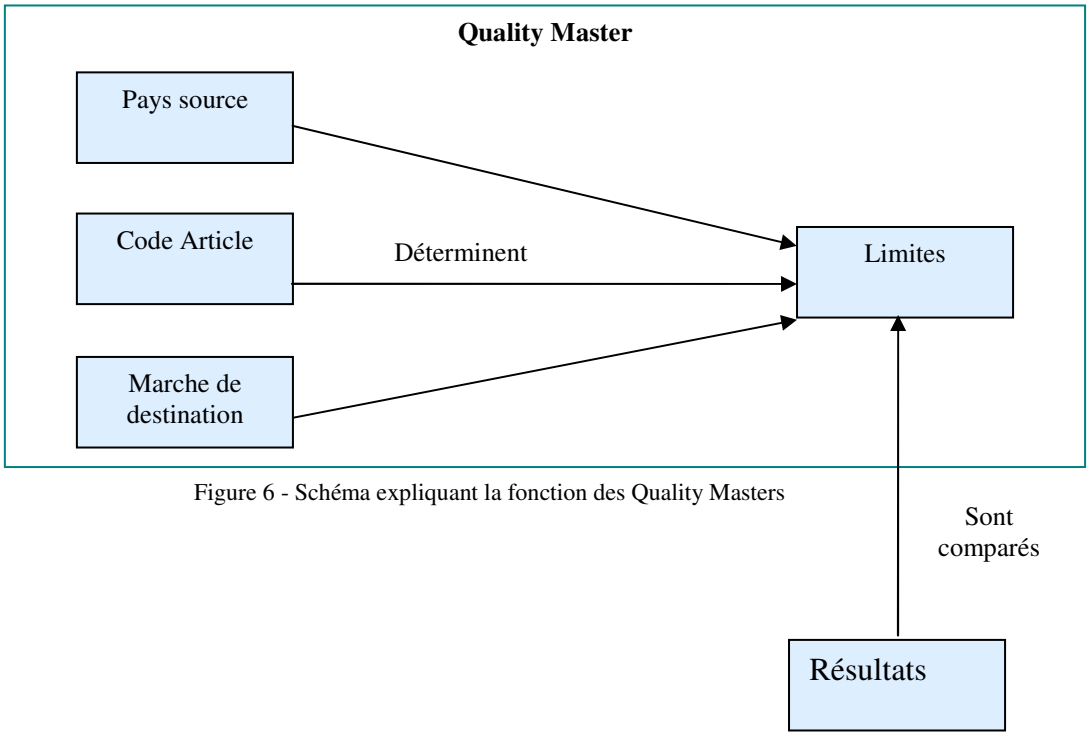


Figure 6 - Schéma expliquant la fonction des Quality Masters

EuroLIMS participe ainsi au cycle d'approbation et de libération des lots de médicaments. Le personnel qualifié revoit les résultats de tests pour les lots produits et approuve ou rejette ceux-ci suite au contrôle qualité effectué.

Afin de pouvoir libérer les lots de produits pour expédition, des certificats d'analyse (‘Certificate of Analysis’) sont délivrés pour chaque lot. Ces documents reprennent des données de fabrication (tests effectuées, composants du produit,…).

③

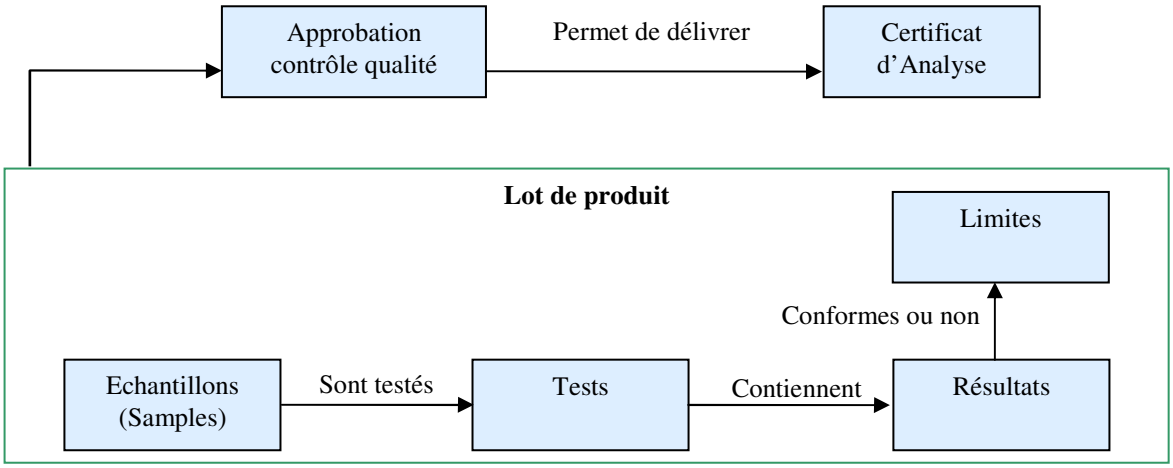


Figure 7 - Schéma de l'approbation d'un lot de produit

En plus de ce flux de données lié à la production d'un lot de produit, on retrouve d'autres types de tests, liés aux données environnementales (l'eau, notamment) ou à la stabilité des médicaments produits.

De plus, une série d'entités données sert à l'organisation des tests. Par exemple, ces derniers peuvent être prévus dans le temps ('Tests schedules'), les échantillons peuvent être créés selon des modèles ('Sample template') et regroupés dans des entités plus larges ('Jobs').

Enfin, EuroLIMS contient des données servant au fonctionnement des modules d'interfaces, de l'application (données de configuration, comptes des utilisateurs et rôles affectés, sécurité…), et aux audits des actions sur les données (insertion, modification, suppression).

Bien d'autres entités de données servent ainsi au fonctionnement d'EuroLIMS.

EuroLIMS est en effet doté d'une base de données de plus de trois cents tables, et la structuration interne des données, dont nous avons vu les fondements, permet de lier de nombreuses tables entre elles. Cette liaison peut se faire soit par les contraintes Oracle (clés étrangères,..), soit par des contraintes mises en place par le code de l'application EuroLIMS.

On le comprend, les développements de l'application BOLDER R3.1 ont dû prendre en compte cette complexité et seule une expertise technique sur l'application EuroLIMS a permis de valider l'organisation des données dans BOLDER R3.1.

I.2.3.3 Une application critique

Le SIGL EuroLIMS, en servant d'application de travail aux équipes laborantines, en déterminant les tests à effectuer et en enregistrant les résultats de ceux-ci, est une application critique pour Lilly.

C'est en effet sur la foi de ces données que Lilly peut s'assurer que ses médicaments sont produits selon les normes pharmaceutiques en vigueur.

Toute action non attendue sur ces données peut donc avoir un impact dit 'SISPQ[11]' sur le produit concerné. L'intégrité de ces données et l'assurance qu'elles soient conservées comme tel dans le temps fait parti du caractère critique d'EuroLIMS.

Dans le cas où un jeu de données de tests enregistré dans EuroLIMS est mis en doute, c'est l'ensemble du lot de médicaments produits qui est remis en cause et peut être envoyé en destruction, causant alors des pertes financières importantes.

I.2.3.4 Volumes d'utilisation d'EuroLIMS

Pour donner un ordre de grandeur de la taille des différentes bases de données des sites, voici un tableau récapitulatif présentant la date de mis en place d'EuroLIMS R1.0, le nombre d'années d'utilisation au 1° mars de cette année, le nombre d'utilisateurs, ainsi que le volume des principales données de base.

BOLDER R3.1 étant développé pour devenir la solution de restitution d'un certain de données historiques d'EuroLIMS, ces chiffres permettent de voir quels seront les volumes de données à transférer et à stocker, pour les données les plus importantes (échantillons, tests, résultats).

Ils seront notamment utilisés lors des estimations de temps de transfert, lors des installations de BOLDER R3.1 pour les instances de production.

[11] SISPQ ou SIPPQ: Sécurité, Intégrité, Principe, Pureté et Qualité. Voir liste des abréviations.

Tableau 3 : récapitulatif des données de base d'EuroLIMS par instance de production

Site	Première utilisation Release 1	Années d'utilisations Release 2 au 1° mars 2012	Nombre d'utilisateurs	Nombre d'échantillons	Nombre de tests	Nombre de résultats
Alcobendas	Février 1996	9	100	126.500	757.000	1.641.500
Basingstoke	Novembre 1995	9	50	49.000	311.500	536.500
Fegersheim	Décembre 1994	10	360	580.000	1.580.000	7.395.000
Giessen	Octobre 2003	8	20	3.000	21.500	52.900
Kinsale	Décembre 1994	10	150	332.000	905.000	2.519.500
Sesto	Février 1996	10	75	72.000	429.000	1.208.000
Speke	Novembre 1998	9	60	101.000	734.500	1.833.000

I.2.4 L'outil de reporting BOLDER

EuroLIMS dispose de ses propres modules de reporting de données, qui sont peu évolués, mais utiles pour un usage immédiat et limité. Les utilisateurs ont manifesté le besoin d'accéder de manière plus fine et plus complexe à un jeu de données bien plus conséquent. L'équipe MICS Lab a donc développé en 1999 une solution de reporting basée sur Business Objects, BOLDER.

Celle-ci permet aux utilisateurs de consulter de manière précise, grâce à des rapports créés selon leurs besoins, les données métier les plus importantes contenues dans EuroLIMS.

L'application comprend un univers et une série de rapports qui sont fournis aux sites, afin qu'ils puissent effectuer des fouilles de données dans la base BOLDER, dont la modélisation a été adaptée du besoin de reporting. Cette base est interfacée avec l'instance de production EuroLIMS correspondante.

Les sites peuvent en plus choisir de développer leur propre univers et leurs propres rapports. On parle alors d'univers et de rapports locaux, en opposition à l'univers et aux rapports globaux fournis par l'équipe Global MI&CS Lab.

I.2.4.1 L'architecture BOLDER

BOLDER est composé de trois éléments principaux, qui reposent sur des technologies éprouvées.

Dans la version 3.0, l'outil d'interface et la base de données sont hébergés sur le même serveur Unix, ceci pour des raisons de facilité de déploiement et de maintenance.

De plus, l'outil d'interface est susceptible d'insérer en une seule fois (transfert complet) un volume de données très conséquent. Il convient donc de limiter la distance entre le serveur de stockage des fichiers plats et la base de données.

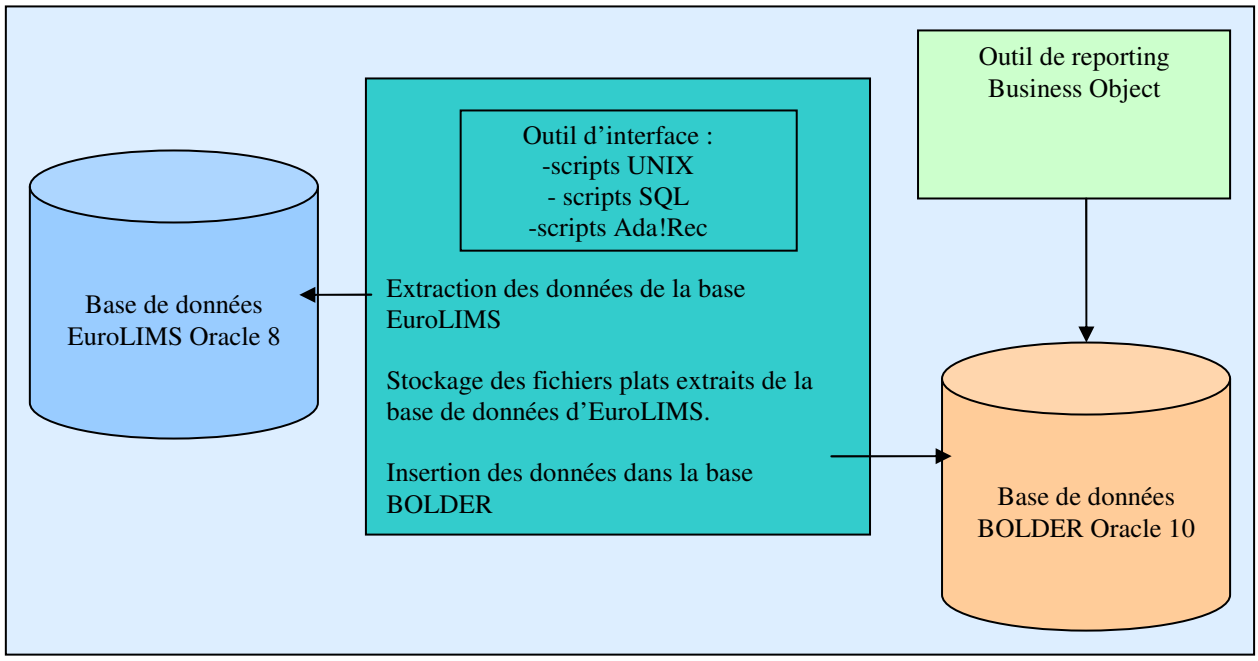


Figure 8 - Schéma d'aperçu de l'infrastructure de BOLDER

I.2.4.2 l'outil d'interface

Cet outil est composé d'un ensemble de scripts Unix Shell, SQL et surtout de fichiers de code de l'application Ada!Rec.

Cet outil, développé par Lilly en 1999, permet d'extraire et d'insérer des données via un transfert de fichiers plats, grâce à des commandes SQL.

Il présente donc aussi quelques possibilités de transformations de données.

Ada!Rec a été utilisé pendant très longtemps par Lilly pour remplir les fonctions d'application ETL[12]et est encore utilisé notamment par certaines interfaces d'EuroLIMS.

L'outil d'interface est démarré et stoppée grâce à une crontab[13] Unix, qui permet de paramétrer avec précision quand doit avoir lieu le transfert de données de la base EuroLIMS vers la base BOLDER. Le transfert est journalier et en semaine pour tous les sites, plus rarement le week-end.

C'est un transfert de type incrémental. Ainsi seules les données insérées, mises à jour ou supprimées depuis le dernier transfert seront reflétées dans BOLDER.

C'est lors de l'installation de l'application BOLDER ou lors d'un besoin spécifique (restauration de données, demande du site), que l'ensemble des données choisies sont transférées d'un bloc d'EuroLIMS vers BOLDER. Et c'est sur cette première couche de données que viendront s'accumuler les autres données transférées de manière incrémentale.

L'outil d'interface contient également des scripts SQL créant les déclencheurs, les vues, les clichés et les droits spécifiques sur la base BOLDER.

On retrouve ci-dessous un schéma montrant comment peut se passer un transfert incrémental pour la table des marchés (liste des pays destinataires des produits) :

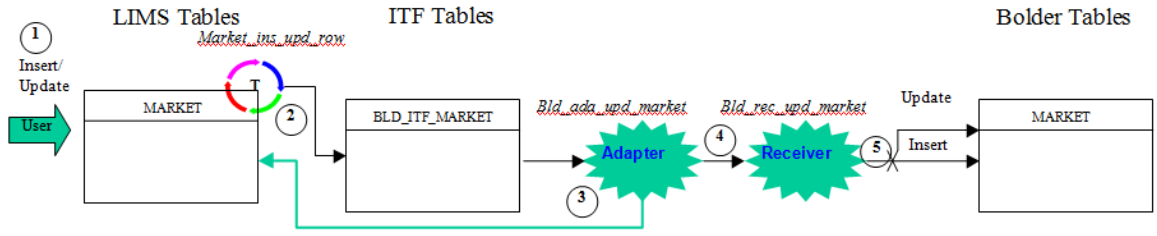


Figure 9 - Schéma explicatif d'un transfert de données incrémental : table des marchés

1) un traitement est appliqué aux données de marchés dans EuroLIMS (insertion, mise à jour).

[12] ETL : Extract, Transform and Load. Voir glossaire
[13] Crontab : commande UNIX ou LINUX permettant de lancer des commandes selon des horaires déterminés.

2) ce qui active un déclencheur (trigger), ici ‘market_ins_upd_row’, qui met à jour la table d’interface correspondante.

3) le script Ada!Rec ‘Adapter’, lancé grâce à la crontab Unix, va, par des commandes SQL simples[14], vérifier via la table d’interface quelles sont les données à sélectionner dans la table des marchés dans EuroLIMS.

4) les données extraites sont stockées dans un fichier plat dans l’arborescence du paquet d’interface sur le serveur Unix.

5) ce fichier plat est pris en charge par le script Ada!Rec ‘Receiver’ qui va agir sur la table des marchés dans BOLDER selon l’action initiée dans EuroLIMS : insertion ou, modification.

Dans le cas d’un transfert global, les étapes 1 et 2 ne sont pas effectuées et le script ‘Adapter’ correspondant va chercher l’ensemble des valeurs des champs dans la table EuroLIMS pour les stocker dans un fichier plat qui sera traité par le fichier ‘Receiver’.

I.2.4.3 La base de données

C’est une base Oracle 10 pour la version 3.0 de BOLDER. Elle contient les données de trente-six tables EuroLIMS (sur plus de trois cents), qui représentent les principales données métier du SIGL.

Comme il s’agit d’une base de données dédiée au reporting, un travail de modélisation a été accompli afin d’agréger une partie des données extraites d’EuroLIMS, de manière à ce qu’elles soient facilement interrogeables dans Business Objects.

Dans la version 3.0, la majorité des données est transférée à l’identique, et peu de tables de BOLDER contiennent des données agrégées.

Nous verrons que la modélisation effectuée pour la version 3.1 a augmenté le nombre de tables agrégées de manière conséquente (plus de soixante tables), et par la même le volume de données à transférer.

[14] Voir Annexe n°6 – script ADA!REC ‘adapter’ pour la table des marchés.

Les tables et les champs BOLDER portent très souvent le nom de la table EuroLIMS correspondante, ceci pour permettre aux utilisateurs de s'y retrouver rapidement dans leurs recherches dans l'univers Business Objects.

I.2.4.4 Business Objects

BOLDER est basé sur la plate-forme de reporting Business Objects. L'application a dû donc suivre les évolutions de cet outil. L'équipe Global MI&CS Lab a ainsi inclus dans les nouvelles versions de BOLDER ce type de migration technique, pour l'univers et pour les rapports.

La version BOLDER R3.0 a ainsi été développée majoritairement pour effectuer la migration de Business Objects 6.5 vers Business Objects XI r2. Et le projet de la version R3.1, objet de ce mémoire, inclut à ce jour la migration vers BOXI r3.

Afin de permettre le test et la validation des éléments Business Objects, trois types d'environnement sont mis à disposition par l'équipe de support : développement, validation et production. Ceux-ci sont connectés à des bases de donnés de type équivalent (développement, validation,…).

I.2.5 La fin de vie d'EuroLIMS et son futur retrait

EuroLIMS est une application robuste et stable, dont le remplacement est engagé depuis plusieurs années. Ce qui s'explique par le fait que Lilly a pris la décision de remplacer ce SIGL par l'application Darwin LIMS, de l'éditeur Thermo Scientific. Ce nouveau SIGL est amené à être utilisé dans tous les sites Lilly nécessitant un LIMS, en remplacement d'EuroLIMS et d'AdminLIMS, son équivalent utilisé en zone 1.

Une des principales raisons est que ce dernier doit être retiré, en raison de son obsolescence. Lilly visant à globaliser une partie de ses applications informatiques, EuroLIMS est donc aussi en cours de remplacement par Darwin.

Un travail d'adaptation et de développements spécifiques de l'application remplaçante Darwin à l'environnement Lilly s'est terminé il y a deux ans. Les déploiements de cette application ont démarré et les sites européens utilisant EuroLIMS ont donc commencé les uns après les autres à l'utiliser.

Il est donc prévu à terme d'arrêter EuroLIMS et de le retirer sur chaque site, six mois après la fin de sa dernière utilisation.

Les données contenues dans sa base de données concernent des lots de médicaments produits, mis sur le marché, vendus et pour beaucoup déjà consommées. Il est donc requis par la loi de les conserver pendant une période d'années définie à la fois par la réglementation du pays où les médicaments sont fabriqués et par les règles internes à Lilly. Cette période peut varier selon le type de données et est comprise entre dix et soixante-dix ans.

Ce projet de retrait implique la création d'outils d'extraction, de stockage et de présentation des données EuroLIMS et a donc amené la mise en place de deux projets :

- un projet d'archivage à court terme. Il s'agit de faire de BOLDER un outil permettant d'extraire, de stocker et de présenter la partie essentielle des données métiers contenues dans EuroLIMS.

- un archivage à long terme de la totalité des données sous forme de fichiers plats, dans un format pérenne. L'intérêt de cet outil est de stocker des données pendant un temps conséquent (plusieurs dizaines d'années), et de pouvoir y accéder de manière simple. Ce projet devrait démarrer dans les prochains mois.

II Deuxième partie : le projet BOLDER R3.1

II.1 Chapitre un : définition du projet

Afin de bien cerner le contenu du projet, nous trouverons dans les lignes qui suivent une présentation détaillée reprenant les objectifs, le cahier des charges, puis les contraintes accompagnant son déroulement.

II.1.1 Enoncé

L'application EuroLIMS utilisée sur sept sites Lilly en Europe va être retirée dans les prochaines années.

Pour des besoins réglementaires, il va falloir archiver une sélection de données de production, afin de pouvoir y effectuer des recherches et les présenter pendant des années lors des inspections réglementaires. Ceci pour pouvoir démontrer comment ont été testés et approuvés les lots de produits, avant mise sur le marché.

La solution informatique servira dans un premier temps d'outil de rapport. Il est en effet convenu par le Global Custodian et les experts technique et métier que la consultation des données d'EuroLIMS se passeraient surtout dans les premières années après le retrait.

Les données devraient donc être consultables de manière aisée pendant la période de transition entre la fin de vie d'EuroLIMS et l'archivage définitif.

Il sera ensuite envisagé si elle deviendra un outil d'archivage et de recherche, une fois l'application EuroLIMS retirée définitivement, ou servira à alimenter l'outil d'archivage définitif.

II.1.2 Objectifs du projet

II.1.2.1 Objectifs techniques et fonctionnels

L'application devra permettre l'archivage d'un ensemble de données d'EuroLIMS prédéfinies dans le cahier des charges. Elle devra pouvoir permettre à des utilisateurs non informaticiens, ayant une connaissance des métiers de laboratoires, de pouvoir rechercher de manière rapide les données archivées.

La solution informatique s'appuiera sur l'application BOLDER R3.0 actuellement en production. Il s'agira de faire évoluer l'outil d'interface, la base de données et l'univers Business Objects, de manière à intégrer un volume de données en provenance d'EuroLIMS doublé.

Les données sélectionnées seront chargées dans BOLDER, ceci pouvant représenter jusqu'à quinze années de données de production. L'application devra donc pouvoir permettre des recherches de données en un temps raisonnable pour les utilisateurs.

Ces données seront présentées au plus proche de leur présentation dans EuroLIMS, en reprenant la catégorisation qui y est faire (exemple : données de stabilités, lots produits, données venant de SAP,…).

Il est aussi demandé à ce que les données soient organisées dans la base de données et l'univers Business Objects de manière à ce que les utilisateurs aient à faire un minimum de liens entre les tables lors de leurs recherches, ceci afin de faciliter l'utilisation de l'application.

L'application sera installée avant le retrait d'EuroLIMS, BOLDER devra donc continuer à jouer son rôle de solution de reporting. Ce qui implique de s'assurer de la non-régression, surtout d'un point de vue performances.

Deux objectifs techniques s'ajoutent : le projet devra également permettre de faire migrer l'application BOLDER vers Business Objects XI r3 et de faire évoluer la base de données vers la version Oracle 11. Ces deux demandes découlent du cycle de vie de l'infrastructure informatique au sein de Lilly.

Enfin la correction de bogues mineurs, découverts lors de l'utilisation de BOLDER R3.0, devra également être effectuée.

II.1.2.2 Objectifs de validation

L'ensemble des artefacts (fichiers de code, rapports Business Objects, documents techniques, tests effectués, documents de validation,…) devront être produits et livrés en respectant les principes de validation en vigueur chez Lilly.

Chaque livrable devra être soumis à une relecture et une approbation par un pair technique et respectera le système de version mis en place.

Les fichiers de code et les éléments permettant le fonctionnement de l'application BOLDER R3.1, créés par l'équipe projet, seront revus et contrôlés selon les procédures de création et de revue de code en vigueur.

Les analystes devront s'attacher à respecter l'identité de chaque instance (développement, test, qualification) en effectuant leurs travaux de développement sur l'instance dédiée afin d'éviter toute confusion.

Si lors des tests informels ou formels, des bogues sont relevés, ils seront corrigés sur l'instance de développement après les tests.

De la même manière, l'instance de qualification doit être considérée comme un reflet d'une instance de production : en aucun cas un débogage ou un changement de configuration pour test n'y est autorisé.

II.1.2.3 Objectif documentaire

Les procédures en vigueur chez Lilly et les contraintes d'un projet informatique imposent une maintenance stricte de la documentation existante.

Les documents de validation doivent être documentés de manière précise et remis à jour dès que l'élément technique correspondant est changé.

L'ensemble des documents maintenus et créés pour la version BOLDER R3.1 devront être listés précisément et comme indiqué revus et approuvés, dans l'ordre du déroulement de

la méthode de projet. Ainsi une revue de code ne pourra être approuvée qu'une fois les spécifications fonctionnelles revues et approuvées officiellement.

Citons parmi ces documents le cahier des charges, l'analyse d'impact de l'application, les spécifications fonctionnelles, les revues de code, les procédures de déploiements, les configurations techniques, le plan de test, les cas de tests, le rapport de tests, la matrice de traçabilité (adéquation entre les besoins du cahier des charges et les tests à effectuer).

II.1.3 Les contraintes externes au projet

Le projet a commencé en février 2011, avec une première contrainte externe forte : le mode de fonctionnement des services informatiques de Lilly est en cours de changement. Ceci a eu comme impact de ne pas pouvoir allouer les ressources ayant les connaissances techniques nécessaires dans les phases clés du projet.

Une autre conséquence fut que peu de personnes ont pu consacrer au projet plus de la moitié de leur temps de travail, les autres ressources intervenant ponctuellement.

Cette première contrainte a été atténuée en octobre 2011, lorsque deux nouvelles ressources ont été intégrées à plein temps au projet.

Une deuxième contrainte externe est apparue au début de l'année 2012. Lilly ayant décidé d'attribuer le contrat de support du Global MI&CS à un nouveau prestataire. L'équipe Global MI&CS Lab à Fegersheim, a ainsi dû participer au transfert de compétences, activités étant plus prioritaires que les activités du projet.

J'ai pour ma part préparé et dispensé un certain nombre de formations, en prenant notamment en charge celles concernant BOLDER R3.0 et R3.1.

Ceci me permettant de confirmer le corpus de connaissances techniques existant et de former les nouveaux analystes qui seront appelés à participer aux tests de BOLDER R3.1

Cette formation devrait donc permettre l'intégration de nouvelles ressources dans l'équipe projet. Mais elle a également ralenti l'avancée des travaux, les experts techniques Lilly et moi-même étant de ce fait moins disponibles.

Enfin, ce projet m'a été confié dans le cadre de la mission que j'effectue au sein de l'équipe Global MI&CS Lab. Celle-ci consiste à assurer la coordination du support et de la maintenance des applications EuroLIMS et BOLDER. J'ai donc dû diriger ce projet en parallèle de mes autres activités, ce qui m'a fréquemment amené à devoir partager mon temps de travail entre le support et le projet, depuis février 2011.

II.1.4 Les contraintes internes au projet

De manière classique, un projet informatique est caractérisé au minimum par trois contraintes : maintien des coûts dans les limites du budget alloué, livraison d'une application répondant au cahier des charges et exempte de bogues (contrainte de qualité), respect des délais.

Dans le cadre de ce projet, je n'ai pas été impliqué dans la gestion des coûts, ce rôle étant dévolu au Global Custodian. Cependant, j'ai dû gérer la contrainte qui a découlé des moyens limités alloués au projet, à savoir des ressources limitées.

II.1.4.1 Contrainte de ressources limitées

Comme indiqué, le projet s'est déroulé dans un cadre de ressources limitées, pour des raisons externes au projet.

L'équipe de projet initiale a ainsi été formée avec des ressources offshores dédiées au support des applications EuroLIMS/BOLDER.

Une progression assez lente a été remarquée relativement vite, du fait de ce nombre de ressources limitées, combiné avec le fait que ces ressources étaient en transition avec la nouvelle société prestataire et l'impossibilité d'accéder aux experts techniques aux moments-clés.

Mon intervention a alors notamment consisté à remonter au Global Custodian l'avancée lente des travaux et mes prévisions sur les travaux futurs, en prenant en compte le rythme réel de développement.

Une fois les travaux de développements commencés en juin 2011, j'ai suivi l'avancée de ceux-ci en réévaluant régulièrement la durée de cette tâche, et la date estimée de fin de projet. Devant ces lenteurs, la décision a été prise par Lilly de renforcer l'équipe projet.

Se sont ainsi rajoutées en septembre 2011 deux ressources dédiées au projet et basées à Fegersheim. Ce renfort a impliqué un certain nombre de tâches de mon côté :

- accueil et intégration des personnes dans l'équipe et l'environnement Lilly

- suivi des formations, nécessaires pour commencer à travailler formellement sur le projet

- accompagnement dans les premiers travaux de développement, afin de s'assurer du bon respect des normes de développement et de validation.

Ces deux ressources sont toujours présentes mais il est actuellement prévu de revenir à une équipe projet étoffée en s'appuyant sur des nouvelles ressources, situées en Inde, à partir de la mi-août 2012.

Celles-ci interviendront notamment dans les phases de tests. Je vais donc prévoir une période d'intégration de ces ressources, afin qu'elles puissent acquérir les méthodes de travail nécessaire pour le testing.

Cela passera soit par une phase de simulation d'exécution des tests, soit par une coordination renforcée de ma part lors des premiers tests exécutés en réel, afin de m'assurer de leurs bonnes exécutions et documentation.

Enfin, l'absence d'expertise sur certains domaines techniques a amené une sous-estimation de certaines tâches lors de la phase de préparation du projet. Celui-ci a en effet démarré sans qu'il soit possible de définir clairement les développements qui auraient lieu dans l'univers Business Objects. L'estimation de temps de développement initial s'est révélée erronée et a dû être réévaluée plusieurs fois, à mesure que les besoins ont été analysés.

J'aborderai ce point plus en détail dans la partie dévolue au plan de projet (Partie deux, chapitre trois).

II.1.4.2 Contrainte de qualité

Le projet de développement s'effectue dans le cadre de validation informatique propre à l'industrie pharmaceutique, pour produire une application d'archivage et de recherche de données laborantines. Nous nous trouvons donc dans un contexte où les contrainte de qualité et réglementaires sont fortes.

Cela implique une méthode de gestion de projet où chaque phase de développement est précédée d'une phase de spécification, et où chaque création est confirmée par une phase de validation.

Cela se traduit donc par une nécessité de test unitaire sur chaque fichier de code développé, mais également une revue des spécifications techniques et du code par un expert technique.

Les tests d'intégration doivent être robustes et doivent permettre de vérifier que le résultat de chaque utilisation (insertion, modification) des données dans EuroLIMS se retrouve dans BOLDER. Les tests les plus complexes doivent de plus être revus et validés par un expert de l'utilisation d'EuroLIMS.

Vu l'ampleur des modifications de ce projet, deux phases de tests ont été prévues dans le plan de déroulement : une première, dite informelle, qui sert à découvrir les bogues et les oublis de développement, une deuxième, dite formelle, qui sert à confirmer leur correction et que le système fonctionne conformément aux requis écrits. Cette phase formelle implique de figer les artefacts créés, en les signant et les approuvant. Toute modification entrainera une nouvelle version et un nouveau cycle d'approbation.

II.1.4.3 Contrainte de temps

La date de livraison de l'application BOLDER R3.1 dépend de deux paramètres : la date du retrait effectif d'EuroLIMS pour tous les sites concernés. BOLDER R3.1 devra avoir été validé et livré avant qu'EuroLIMS ait cessé de fonctionner. Cette date de fin d'activité est prévue pour fin 2014.

La mise en production de BOLDER R3.1 sur les sites utilisateurs permettra de s'assurer qu'elle répond correctement aux besoins d'archivage des données des sites.

Le deuxième critère est d'ordre technique : BOLDER R3.0 fonctionne avec la plate-forme Business Objects XI r2. Celle-ci devait être remplacée par la version suivante (BOXI r3) initialement avant juin 2011. Cependant, cette date a été renégociée plusieurs fois, notamment en raison du temps de développement de BOLDER R3.1.

Notons enfin que cette contrainte de temps s'est retrouvée fortement renforcée à la fin du mois de juin 2012. Certaines des données contenues dans BOLDER R3.1 sont devenues primordiales dans le cadre du développement d'un entrepôt de données multi-applications. La livraison de la version BOLDER R3.1 est donc devenue plus critique et il m'a été demandé d'organiser les actions restantes de manière à ce que le projet progresse plus vite, pour délivrer si possible avant la fin du premier trimestre 2013.

II.2 Chapitre deux : la mise en place du projet

II.2.1 Le recueil des besoins

L'équipe Global MI&CS Lab a commencé à rassembler les besoins des sites utilisateurs en octobre 2010. Pour ce faire, une liste des données à transférer à été élaborée, en rattachant chacune d'entre elles à son contexte, au menu permettant d'y accéder et à son écran de saisie. Cette liste a été transmise aux Power Users[15] desdits sites, à charge pour eux de confirmer ou d'infirmer la nécessité d'accéder à cette donnée dans BOLDER, après le retrait d'EuroLIMS.

Cette méthode s'explique de par la manière dont a été construit dès le départ le cahier des charges (ou 'requirements' en anglais) de la première version BOLDER. Ce document contient l'ensemble des requis de l'application depuis le début. Il est basé sur une approche majoritairement technique.

Il présente donc notamment une liste des données à transmettre dans BOLDER[16], assortie du besoin de refléter les traitements de celles-ci dans EuroLIMS (insertion, mise à jour, suppression). Une série de requis fonctionnels concerne les rapports Business Objects livrés avec BOLDER.

Ce recueil de besoins a été complété par des requis techniques liés aux changements d'infrastructures (par exemple : migration des bases de données vers Oracle 11, migration vers BOXI r3) et des corrections de bogues mineurs détectés lors de l'utilisation de la version R3.0.

La réunion du lancement du projet a pu ainsi avoir lieu le 3 février 2011, une fois une grande partie des besoins rassemblés, mais tous les retours des sites utilisateurs n'ont pu être finalisés que trois mois plus tard.

Depuis le début du projet, deux nouvelles versions du cahier des charges ont été créées, avec des compléments d'informations. On trouvera ci-dessous un récapitulatif des requis du projet, répartis par type.

[15] Power User : utilisateur expert d'une application. Voir glossaire.
[16] Voir extrait du cahier des charges dans les annexes 1 et 2

Tableau 4: récapitulatif des requis du cahier des charges, classés par type

Module	Description	Nombre de requis
SYS	Requis généraux concernant le système	7
ERS	Enregistrements et signature électroniques	8
TRN	Transferts de données entre les bases EuroLIMS et BOLDER	99
SEC	Requis de sécurité (intégrité de données, …)	12
DAT	Restitution de données	6
REP	Requis généraux pour tous les rapports standards	5
STR	Requis concernant les rapports globaux fournis avec BOLDER	32
Total		169

On retrouve dans ce tableau la principale raison d'être de BOLDER R3.1 : le transfert de données d'EuroLIMS et leur affichage dans Business Objects. Le nombre de requis s'y rapportant le montre clairement : 99 requis de transferts (TRN) sur un total de 169 requis.

Viennent ensuite les requis relatifs au jeu de rapports fournis avec l'application (32 requis STR) : le changement de la structure de BOLDER, tant d'un point de vue des données que d'un point de vue de l'infrastructure, imposent de prendre en considération ces requis et de tester ces rapports, a minima en non-régression[17]. Ces requis sont issus du cahier des charges de la version R3.0 et ne représentent pas de nouveaux besoins exprimés.

Ces rapports devront être modifiés dans le cas où ils ne se révèlent plus adaptés à la nouvelle structure de BOLDER.

On remarque par contre que très peu de nouveaux requis concernent les artefacts Business Objects, qui se trouve pourtant être l'outil de restitution des données, avec des développements parfois complexes.

J'ai remonté ce point au Global Custodian, en arguant que le nombre limité de tests d'intégration pour cette partie rendait plus délicat la validation des développements effectués. Cet état de fait a été acté par le client Lilly, et il devra être ainsi documenté dans les spécifications fonctionnelles une série de requis concernant Business Objects. Les tests fonctionnels servant à couvrir ces requis restent encore à définir, avec l'aide de l'expert métier.

[17] Le test de non-régression consiste à s'assurer que les changements apportés à l'application n'impactent pas des éléments non-modifiés par ailleurs.

II.2.2 La méthode de projet

Tout projet informatique nécessite une méthode de projet, adaptée aux circonstances et au contexte afin de mener ledit projet à son terme de manière satisfaisante.

Il existe un certain nombre méthodes de projets informatiques disponibles, dont la structuration pourrait s'adapter au projet BOLDER R3.1. Cependant, les procédures en vigueur au sein du Global MI&CS Lab imposent l'utilisation d'une méthode de projet créée par l'équipe et améliorée au fil du temps.

Par conséquent, je n'ai pas eu à devoir faire une analyse détaillée des méthodes de projet existantes et à choisir parmi elles laquelle serait la adaptée à la conduite du projet.

J'ai cependant dans les lignes qui suivent fait l'analyse des caractéristiques du projet BOLDER R3.1, et ai envisagé quelle type de méthode j'aurais pu employer, dans le cas d'une liberté de choix.

II.2.2.1 Etat des lieux

- **Un projet de taille moyenne**

Le projet BOLDER R3.1 peut être considéré comme étant de taille moyenne, et a fait intervenir jusqu'à présent trois à cinq ressources selon les phases.

Il concerne l'évolution d'une seule application, notamment la structuration des données et leur présentation. En ce qui concerne sa durée, les contraintes externes font que son développement s'étale dans le temps, sur plus d'un an.

Ceci implique que la méthode de projet doit pouvoir permettre la gestion d'un cycle de livraison relativement long, avec des ressources qui peuvent varier dans le temps et dans l'espace géographique.

- **Une gestion de données critiques**

Comme nous l'avons vu, BOLDER R3.1 servira d'outil de rapport, puis d'outil d'archivage et de recherche de données laborantines. Ceci pour pouvoir démontrer lors des

inspections réglementaires comment ont été testés et approuvés les lots de produits, avant mise sur le marché.

De ce fait, la criticité des données EuroLIMS ne permet pas un développement rapide. En effet, le code doit être robuste et la validation de la nouvelle version de BOLDER doit pouvoir confirmer qu'elle présente bien les données de la manière dont elles sont traitées dans EuroLIMS.

De plus, chaque site ayant pu utiliser le système de manière particulière, l'ensemble des fichiers de code doit être testé unitairement sur une copie de chaque instance de production.

- **Un cycle de validation conséquent**

Le processus de validation d'applications au sein de Lilly impose une formalisation forte, qui demande pour chaque document des phases de revue et d'approbation, avec un ordre d'approbation à respecter formellement.

Il n'est donc pas concevable de livrer BOLDER R3.1 en plusieurs étapes, de manière itérative : les activités de validation et de déploiement pour l'équipe Global MI&CS Lab et pour les sites imposent une charge de travail conséquente. Il est donc difficile de réaliser des développements et des livraisons itératives rapides sur les instances de production.

Enfin, l'évaluation des besoins des futurs utilisateurs s'est faite en une seule fois. L'ensemble des données à transférer a été validée par les représentants de six sites, et on imagine mal réitérer cette action sans surcoût conséquent.

Le projet BOLDER R3.1 requiert donc une méthode de projet éprouvée, qui permette de s'assurer que l'application produite réponde bien aux besoins des utilisateurs dès la première livraison, et qui comprenne donc des phases de modélisation des traitements des données et de validation de la robustesse du code.

- **Aperçu des méthodes de projet existantes**

Je classe ici les méthodes de projet en deux grandes familles, avec une distinction basée sur le rythme de livraison et la taille du projet.

- les méthodes ‘classiques’, qui proposent un modèle stable et formel, avec des séquences bien définies : cycle en V, cycle en W, modèle en spirale,…

Ce sont des méthodes qui prônent un cycle de développement conséquent, avec comme point de départ une version définitive du cahier des charges. Les différentes phases s’enchaînent formellement les unes après les autres et la mise en production se fait une fois le cycle complet effectué.

Le formalisme que ces méthodes demandent peut se transformer en lourdeur, surtout pour des projets de taille modeste.

Par contre, elles assurent un développement du projet stable et continu, ce qui permet de travailler efficacement dans la durée.

- les méthodes dites ‘agiles’ : elles sont la réponse en contre-point aux méthodes ‘classiques’.

On pense notamment à Scrum, Rapid Application Development ou Crystal Clear.

Mises en avant par le Manifeste Agile, ces méthodes prônent un développement du projet plus proche des utilisateurs, en répondant rapidement à leurs besoins et en revenant vers eux régulièrement. Le cycle de développement se fait de manière itérative, afin de pouvoir livrer fréquemment de nouvelles versions intermédiaires de l’application.

Les membres de l’équipe projet sont idéalement dans la même pièce de travail, afin de stimuler la réflexion technique en commun.

Enfin, l’aspect documentaire est souvent le parent pauvre. Le rythme de travail, le besoin de revenir régulièrement vers l’utilisateur implique que la documentation ne soit pas forcément détaillée, afin d’éviter les mises à jour trop lourdes.

Au vu de ces informations et des caractéristiques du projet BOLDER R3.1, mon choix se serait porté vers une méthode ‘classique’, comme le cycle en V, qui permet un développement sur un rythme stable, avec une documentation détaillée et une validation à partir d’une seule version figée du cahier des charges.

Mais comme indiqué, il existe une méthode de projet dans le corpus de formations l’équipe, qui est imposée pour la gestion de projets.

II.2.2.2 La méthode de projet du Global MI&CS Lab

- **Une méthode construite dans le temps**

Comme pour de nombreuses activités, le développement et la validation informatiques sont conduites chez Lilly selon des méthodes élaborées et améliorées par les équipes projet au fil du temps.

Ces équipes sont souvent composées des personnes de l'équipe en charge d'applications, et qui sont en poste pendant plusieurs années. Ce qui permet une réflexion approfondie sur l'amélioration et l'adaptation des méthodes et procédures utilisées lors de projets.

En ce qui concerne l'équipe Global MI&CS Lab, la méthode de projet fait partie des procédures en vigueur au sein de l'équipe. Elle s'intègre dans le cadre des contraintes réglementaires de validation.

Elle présente un cycle de projet informatique, comprenant un séquençage des actions à effectuer, de l'écriture du cahier des charges par l'équipe projet, à la gestion des tests, en passant par la livraison des différents artefacts, et des possibles itérations sur des éléments bien distincts.

La méthode se focalise sur la partie du déroulé des tests, du traitement des tests échoués, et des corrections à apporter. Cette focalisation est rendue nécessaire par le besoin de s'assurer d'une validation informatique maîtrisée.

Cette méthode présente enfin les différents documents de validation à livrer, étapes nécessaires dans le cycle de projet.

Nous observons cependant une plus grande latitude quant aux méthodes employées pour le recueil des besoins et le développement.

L'idée est de permettre une souplesse et permettre d'avancer en parallèle sur différents types de travaux, par exemple le développement et l'écriture des tests, tout en gardant un formalisme nécessaire.

Celui-ci sert de garde-fou à l'ensemble des activités de spécification d'écriture et de développement, en imposant un séquençage bien précis de ces étapes.

Ainsi, les spécifications fonctionnelles doivent être approuvées après l'approbation du cahier des charges, les revues de code étant réalisées après l'approbation des spécifications. Ceci demande ainsi à l équipe projet de s'assurer que les artefacts sont validés dans un ordre cohérent.

- **Présentation schématique**

Pour une présentation plus claire, j'ai découpé le schéma explicatif dudit cadre en plusieurs parties distinctes, afin de rendre le schéma directeur plus visible. On retrouvera l'entièreté du schéma en annexe[18].

Les explications du schéma sont en anglais, langue de travail au sein de l'équipe Global MI&CS Lab.

La méthode présente d'abord une phase de démarrage et développement dans laquelle on retrouve les premières étapes de toute méthode de projet :

- cahier des charges (n°1-'Requirements'),

- spécifications fonctionnelles (n°2-Design),

- écriture du code (n°3),

- identification des cas de tests à effectuer (n°4).

18 Voir Annexe n°3 « Méthode de gestion de projet en vigueur au sein du MI&CS »

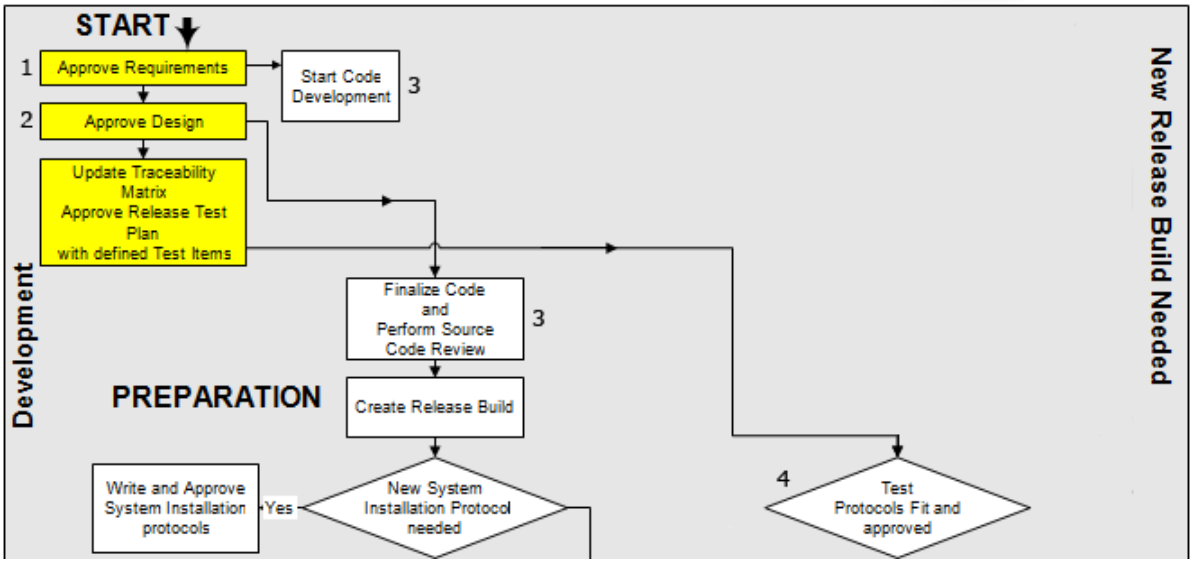


Figure 10 - Méthode de projet du Global MI&CS Lab : étapes initiales du projet

Puis vient une phase dédiée à l'installation de l'application et à la mise en place des données de tests (partie gauche).

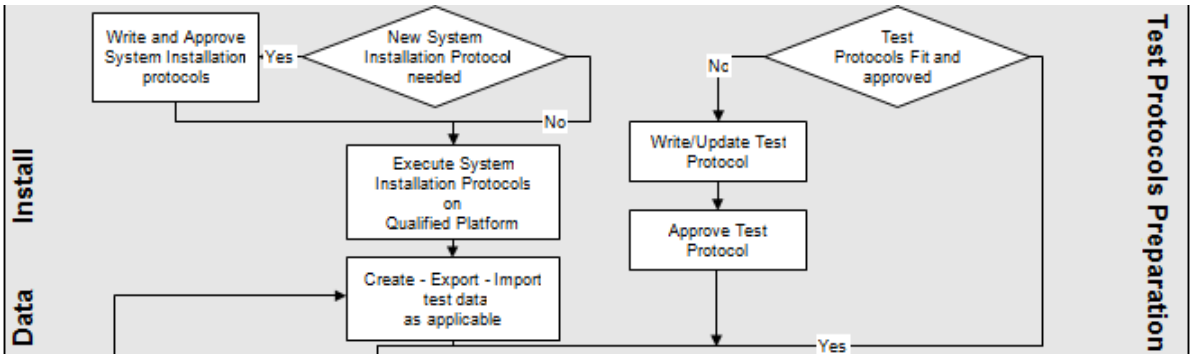


Figure 11 - Méthode de projet du Global MI&CS Lab : installation et écriture des tests

La phase d'écriture et d'approbation des tests à exécuter (partie droite) peut s'effectuer en parallèle de cette phase d'installation, mais ceux-ci doivent être approuvés après les spécifications fonctionnelles, et avant le commencement de la phase de test.

Enfin nous retrouvons la phase d'exécution des tests. Comme annoncé, elle est beaucoup plus détaillée que les premières phases.

On observe dans le schéma ci-dessous comment s'organise le déroulé des tests, notamment comment sont traités les tests échoués, comment on doit gérer les bogues découverts, et enfin comment on aboutit à une phase de tests complète.

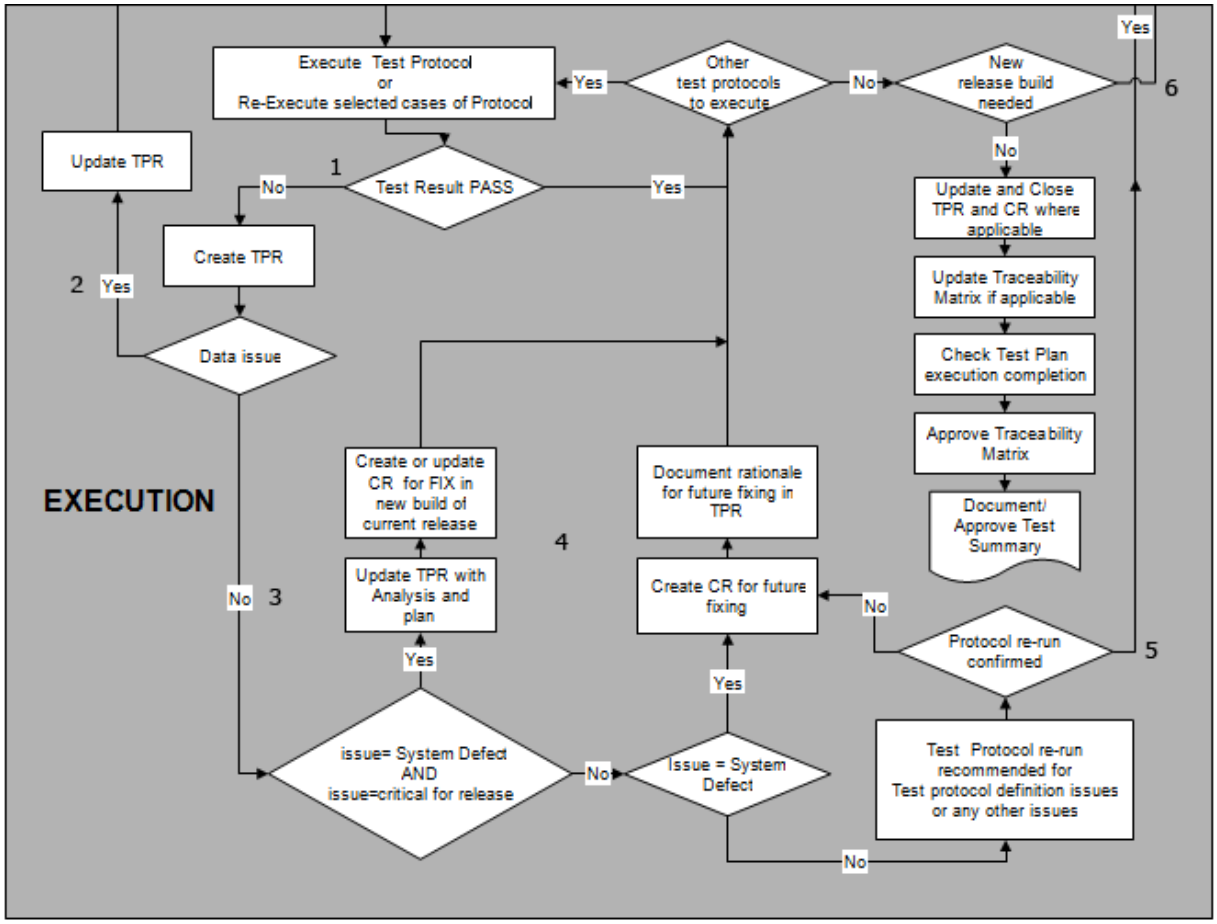


Figure 12 - Méthode de projet du Global MI&CS Lab : étapes à respecter lors des tests

En cas de test échoué, une investigation est ouverte (un TPR [19]) (1, figure 12)

Dans le cas d'un problème avec le jeu de données de test (2, figure 12), on utilise un nouveau jeu de données et on exécute à nouveau le test :

[19] TPR : Test Problem Report. Il s'agit de l'investigation et de la résolution d'un problème rencontré lors d'un test exécuté.

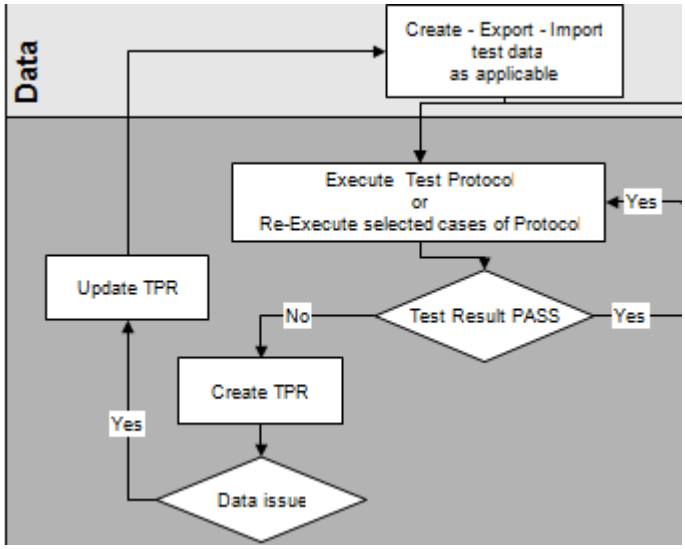


Figure 13 - Méthode de projet du Global MI&CS Lab :
cas d'un test échoué pour cause de données

Si la cause n'est pas due à un problème avec le jeu de données de tests, il faut déterminer si le problème est dû à un bogue (3, figure 12).

En ce cas, une 'demande de changement' (Change Request - CR[20]) est créée pour tracer la modification technique à apporter (4, figure 12).

Il est aussi possible que le problème vienne soit de la conception du protocole de test ou de son exécution : celui-ci doit éventuellement être exécuté à nouveau (5, figure 12), dans le cas où il est statué que le problème ne vienne pas d'une mauvaise interprétation.

En cas de besoin, afin de corriger les bogues détectés une nouvelle itération de développement et de test peut être décidée, demandant de repartir de l'étape de développement (6, figure 12).

[20] CR : Change Request. C'est l'appellation donnée aux demandes de changements informatiques dans une application.

Nous retrouvons alors la première partie de la méthode de projet, incluant cette fois-ci les étapes de mise à jour éventuelle du cahier des charges et des schémas de spécification techniques, avant d'identifier les cas de tests à effectuer à nouveau.

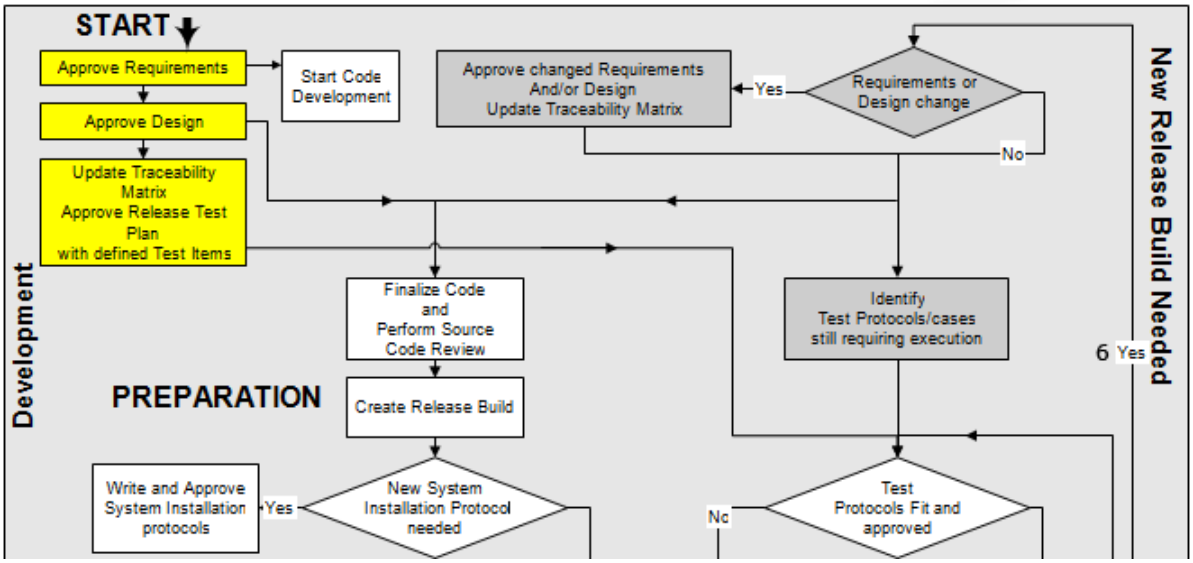


Figure 14 - Méthode de projet du Global MI&CS Lab : nouvelle itération de tests

Pour conclure le testing, la méthode de projet demande à ce que tous les tests échoués soient à nouveau exécutés, avec succès, et/ou que les bogues rencontrés soient analysés, pour une future correction. La documentation de validation est alors mise à jour.

- **Artefacts**

Comme on le voit dans ce schéma, la méthode de projet présente une série d'artefacts, dont l'approbation détermine l'avancement du projet :

- le cahier des charges ('Requirements') : il recueille les besoins des utilisateurs, les contraintes techniques propres à l'application (performances, sécurité) et celles liées aux composants techniques (version de la base de données,…).

- les documents de spécification ('Design documents') : ils présentent les mécanismes techniques de chaque élément de l'application.

- les revues de code ('Source Code Review') : il s'agit de revues formelles, effectuées sur chaque fichier de code de l'application. Son but est d'assurer le respect des standards de

développement et que chaque fichier a été testé unitairement. Chaque revue doit être signée comme acceptée pour que le fichier de code soit intégré dans le paquet d'installation.

- les cas de tests : il s'agit pour la version BOLDER R3.1 de cas de tests existants qui ont été remaniés, ou de nouveaux cas de tests qui ont été écrits pour tester les nouvelles fonctionnalités de l'application, les corrections apportées ou les nouveaux composants techniques.

- la matrice de traçabilité ('Traceability Matrix') : elle permet d'assurer que chaque requis, issue du cahier des charges, a été testé.

- le plan de test ('Test Plan') : il détermine la stratégie de test, les tests qui vont être effectués.

- les protocoles d'installation : ces documents tracent de manière précise les étapes d'installation de l'application et de ses composants techniques. L'exécution correcte de ces protocoles assure que l'application est déployée de la même manière sur toutes les instances de qualification et de production. Ceci permet d'assurer un support et une maintenance homogènes.

- le rapport de test ('Test Summary Report') : il résume l'ensemble des tests effectués, les bogues rencontrés, justifie les éventuels écarts au plan défini et la conclusion générale des tests.

Deux documents encadrent ce groupe d'artefacts et représentent les étapes d'initiation et clôture du projet :

- le document d'analyse d'impact de l'application ('Release Impact Assessment').

Il liste tous les objectifs du projet, les impacts que le développement, la validation et la livraison de l'application va avoir sur la documentation de validation, l'ensemble des fichiers de code et l'architecture de l'application. Il présente également le plan de validation qui va servir à prouver que l'ensemble des requis exprimés ont été développés, testés et sont fonctionnels.

- le document de description de l'application ('Release Description Document') : il décrit l'ensemble des artefacts livrés, les éventuels requis non développés ; les écarts avec les prévisions faites dans le document d'analyse d'impact, et évalue de manière synthétique la validité de l'application livrée.

- **Avantages par rapport à d'autres méthodes**

La méthode de projet du Global MI&CS Lab présente trois avantages par rapport aux autres méthodes de projet :

- elle est issue de la réflexion et de l'expérience des équipes informatiques qui l'ont utilisée au fil du temps. Les leçons tirées lors des projets informatiques en milieu pharmaceutique ont permis de l'adapter aux besoins de validation, en gardant les éléments de base de toute méthode de gestion de projet (cahier des charges, spécifications, …). D'autres méthodes de projet plus théoriques seraient sûrement à adapter au contexte pharmaceutique, avant de commencer le projet.

- c'est une méthode qui intègre la gestion de la documentation de validation requise en milieu pharmaceutique et qui en fait des étapes décisives. L'approbation de documents de validation demandés par la réglementation de Lilly fait partie du cycle de vie du projet.

- enfin, c'est une méthode qui doit être connue de chaque ressource de l'équipe projet, en tant que membre de l'équipe Global MI&CS Lab. Elle fait partie du cursus de formations que chaque membre doit suivre, ce qui assure une uniformité de connaissances.

Cette méthode présente donc un certain nombre d'avantages par rapport aux autres méthodes et elle s'est aussi imposée car faisant partie des procédures en vigueur au sein du Global MI&CS Lab. Cette méthode a donc été utilisée pour le projet BOLDER R3.1.

Comme il a été décrit, cette méthode permet d'être souple dans l'ordonnancement des tâches du projet, tout en gardant la rigueur nécessaire aux contraintes de la validation informatique en milieu pharmaceutique. Nous avons donc pu entamer les travaux de développement après une première confirmation du schéma de la base de données, ceci afin d'optimiser l'utilisation des ressources et de limiter les temps de développements.

Cette même démarche s'est poursuivie avec le développement des interfaces, où l'écriture du code et du design permet de tester immédiatement le modèle produit.

Afin de pallier au risque d'un développement erroné, il a de plus été ainsi prévu de revoir de manière itérative avec l'architecte technique les modèles et le codes créés afin de s'assurer de la cohérence des schémas de spécifications techniques.

II.2.3 Le périmètre des responsabilités du chef de projet

Dans la mesure où les responsabilités de direction de l'équipe Global MI&CS Lab complètent les responsabilités de chef de projet, je vais préciser le périmètre de celles-ci, ainsi que leur évolution tout au long du projet.

En tant que chef de projet, j'ai reçu la responsabilité de créer et gérer le plan de projet, de déterminer les tâches à effectuer en fonction des ressources allouées au projet.

Cependant je n'ai pas participé au choix de ces ressources, ceci relevant du chef de service et je n'ai pas non plus pu attribuer de tâches aux ressources non permanentes sans en discuter avec le chef de service, notamment pour les employés Lilly.

Enfin, BOLDER R3.0 reposant sur la plate-forme Business Objects XI r2, et celle-ci devant avoir été retirée depuis de nombreux mois, la renégociation de sa prolongation a incombée au Global Custodian de BOLDER.

On trouvera ci-dessous un tableau présentant le champ de responsabilités qui m'ont été attribuées et celles du directeur de service (Global Custodian).

Celui-ci par ailleurs a représenté les intérêts du client Lilly, en statuant notamment sur le niveau acceptable de qualité de validation, ou sur l'acceptation d'une date de livraison imprécise du fait du manque de ressources.

Tableau 5 : récapitulatif des responsabilités pour le projet BOLDER R3.1

Champ de responsabilité	Chef de projet	Global Custodian (chef de service Global MI&CS Lab)
Ressources	Attribution des tâches aux ressources	Allocation des ressources au projet
Estimation des temps de développement	En collaboration avec des ressources expertes	Première estimation, en début de projet, pour allocation des ressources
Planification	Gestion du plan de projet et escalade des retards et difficultés	Décision et ordonnancement vis-à-vis des contraintes externes
Analyse d'impact	Evaluation des artefacts de BOLDER	Intégration du projet dans la stratégie Lilly
Relation avec équipes infrastructures	Coordination technique	Négociation des contrats de support
Testing	Etude du nouvel outil, migration, écriture, revue	Approbation de l'outil de test
Stratégie de test	Tests et validation planifiés	Approbation de la stratégie de test
Gestion de l'avancée des travaux	Suivi quotidien, support aux analystes, gestion des artefacts	Suivi hebdomadaire de l'avancée du projet
Suivi technique	Revue des artefacts, demandes de modifications	Décisions stratégiques pour les développements à reporter

Il est à noter également, qu'en plus des tâches de planification, d'analyse d'impact, d'inventaire documentaire et de coordination technique, qui ont été dès le départ de mon unique responsabilité, d'autres parties m'ont été confiées de manière progressive : par exemple l'allocation des tâches aux ressources, la planification, ou l'estimation des temps de développement. J'ai cependant remonté quand il l'a été nécessaire les besoins d'expertise technique au Global Custodian, pour assurer une revue des artefacts produits, par l'architecte technique.

Le choix des tâches à effectuer, ainsi que leur attribution, m'ont été complètement confiées une fois la modélisation de la base de données cible confirmée par l'architecte d'EuroLIMS, fin juin 2011.

Cette décision s'explique par le fait que le modèle de base de données a été fournie par l'équipe offshore, et que le chef de service Lilly souhaitait s'assurer que le projet démarre sur des bases solides et validées, avant de me laisser prendre la direction complète du projet.

II.2.4 L'équipe des débuts

Elle a été constituée d'un noyau de deux ressources, auxquelles se sont ajoutées selon les besoins des membres de l'équipe Global MI&CS Lab, afin d'apporter de manière ponctuelle leur expertise technique. On compte parmi ces deux ressources : le chef de projet (moi-même) et un développeur-analyste, rôle ayant été tenu par différentes ressources durant le déroulement du projet.

En effet, le contrat de prestation de support entre Lilly et le prestataire offshore a imposé la mise à disposition de ressources formées, sans qu'il soit nécessaire que ce soit toujours les mêmes personnes.

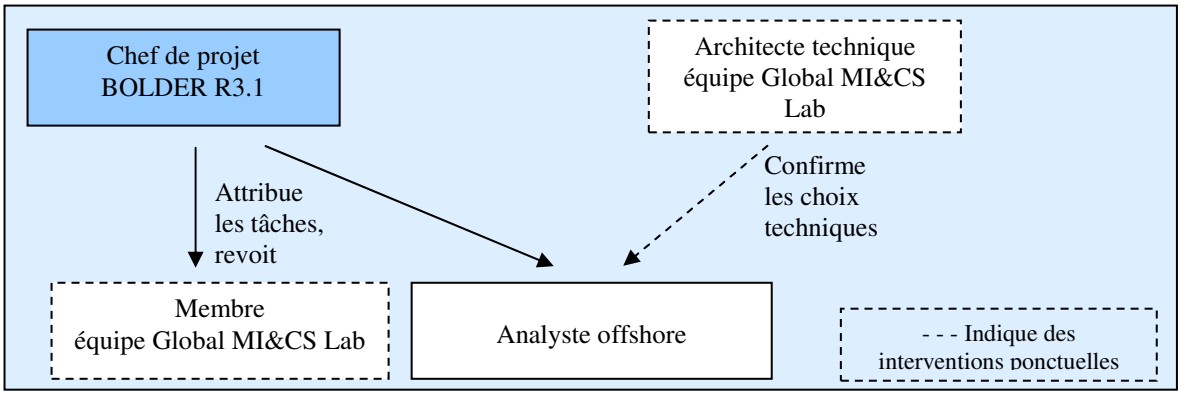


Figure 15 - Schéma équipe projet des débuts

II.2.5 Un démarrage assez lent

Le retrait d'EuroLIMS est prévu depuis plusieurs années. Par contre, c'est un projet non prioritaire par rapport aux autres activités en cours dans l'équipe.

Ce qui implique une incertitude quant à la date réelle de besoin de la livraison de BOLDER R3.1 mais aussi peu de ressources assignées telles que demandées, et donc une lenteur dans le démarrage du projet. L'équipe projet initiale n'a été constituée que de personnes venant de l'équipe support.

En conséquence, aucune nouvelle ressource n'a été intégrée à l'équipe, et formée, avant les six premiers mois de travaux.

Seules deux ressources ont été attribuées à temps plein au projet au bout de cette période, des membres Lilly de l'équipe venant apporter ponctuellement leur expertise.

J'ai pris en compte ce rythme de progrès du projet, pour mettre à jour le plan de projet et estimer ainsi sa date de livraison. Celle-ci a reculée dans le temps au fur et à mesure de l'avancée du projet et j'ai donc alerté le Global Custodian de ce fait.

II.2.6 L'équipe renforcée

A partir du mois de septembre 2011, deux nouvelles ressources basées à Fegersheim sont donc venues rejoindre l'équipe projet à plein temps.

Ce renforcement a été décidé par le client Lilly, pour partiellement compenser l'impact de la phase d'intégration du nouveau prestataire, pendant les travaux de développement du projet.

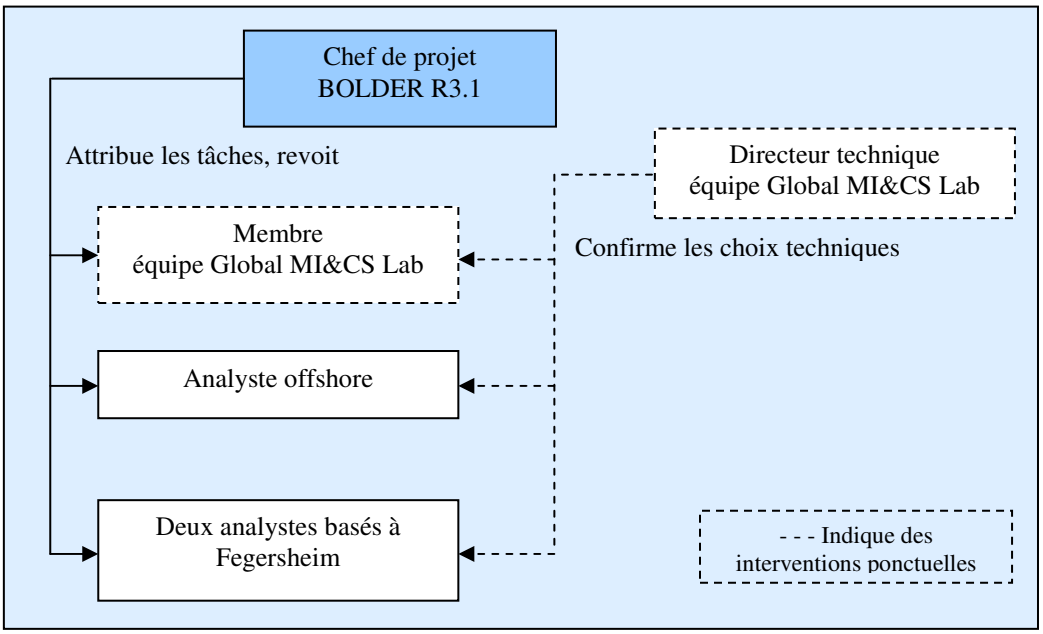


Figure 16 - Schéma équipe projet renforcée

Il est à noter qu'à partir du mois de juin 2012, les travaux de projet se sont déroulés avec une équipe réduite, du fait de l'arrêt du contrat de service du prestataire offshore sur le départ. L'équipe projet a alors été composée du chef de projet et des deux analystes basés à Fegersheim.

II.2.7 La gestion des priorités de la mission

II.2.7.1 Temps partagé entre le support et le mode projet

En raison de l'organisation de l'équipe Global MI&CS Lab et du temps de formation à EuroLIMS, je suis seul à assurer le rôle de coordinateur de support.

Depuis le début du projet, mon temps de travail s'est partagé entre ce rôle de coordinateur et celui de chef de projet.

J'ai donc dû faire preuve de souplesse pour passer d'une tâche de réflexion et de recherche d'informations (exemple, lors d'une revue de fichiers de code), à une tâche d'analyse et de réaction rapides afin de faire face à un incident de production.

Le fait d'avoir été depuis près de trois ans membre de l'équipe support EuroLIMS/BOLDER, et son seul coordinateur depuis plus de deux ans, m'a permis d'acquérir la souplesse et la réactivité requises, pour passer d'une activité à l'autre.

Cette différence de modes de fonctionnement a été un élément à prendre en compte lors de l'augmentation du nombre de ressources allouées et donc des tâches en cours dans le cadre du projet.

Mon attention a dû se répartir sur un nombre grandissant de tâches, afin de revoir et analyser les nombreux livrables, et demander de nouvelles modifications.

Pour faire face à cette situation, j'ai choisi dans un premier temps d'organiser mes journées de travail, en fonction des actions de support, utilisant le temps restant pour travailler sur le projet BOLDER, mais également collaborer avec les analystes de l'équipe projet.

Les activités de formation données au nouveau prestataire au printemps 2012 ont impacté mon temps de disponibilité et j'ai dû compter sur l'autonomie des analystes projet pour faire avancer les développements.

En juin 2012, les activités de support se sont par la suite ralenties et j'ai pu consacrer la quasi-totalité de mes journées au projet.

II.2.7.2 Favoriser l'autonomie des analystes

Le support des instances de production étant prioritaire, j'ai rapidement choisi d'encourager les analystes de l'équipe projet à travailler de manière autonome et à leur confier la responsabilité d'un domaine de développement : la migration des tests, le développement de l'univers Business Objects, ou encore le développement de l'outil d'interface. Cette confiance et cette autonomie accordées correspondent par ailleurs au mode de fonctionnement en vigueur chez Lilly, où une grande confiance est accordée aux collaborateurs dans l'exécution de leurs tâches. Ce qui encourage la prise d'initiatives et de responsabilités.

L'attribution des tâches du projet s'est faite de manière à ce que leur réalisation nécessite peu d'interventions de ma part pendant leur déroulé, une fois l'intégration des ressources effectuées. Mon intervention en tant que chef de projet n'a pas été technique, sauf pour des cas ponctuels, pour lesquels j'ai apporté mon aide et mon expérience, ou pour lesquels j'ai pris des décisions de développement, après consultation des analystes.

Je me suis ainsi consacré à définir avec les analystes l'attendu (exemple : fonctionnalités du code, contenu d'un document de validation), puis me suis appuyé sur leur autonomie pour la réalisation en elle-même, avant de contrôler et valider avec eux ce qui a été développé. J'ai pris en charge la réception des fichiers de code et ai créé le paquet d'installation de l'application, afin de procéder régulièrement à des tests unitaires complets.

De plus, l'outil d'interface étant paramétré pour envoyer des messages d'erreur en cas d'extraction ou d'insertion de données échouées, j'ai pu être tenu informé de ces erreurs et ai alors demandé aux analystes un état de la situation technique.

En cas de besoin, nous avons sollicité l'architecte technique pour qu'il confirme les choix de développement sur les points les plus ardus.

Enfin, j'ai organisé des points réguliers au sein de l'équipe projet et avec le Global Custodian, afin de communiquer et d'échanger sur l'avancée du projet, sur les points bloquants et sur les besoins d'expertise nécessaires. J'ai délibérément choisi de laisser les analystes présenter leurs travaux et les résultats qu'ils ont obtenu, partant du principe qu'ils connaissaient mieux que moi leur sujet, et que le mérite du travail accompli leur revenait en premier.

II.3 Chapitre trois : l'écriture du plan de projet

Tout projet informatique est constitué de phases, quelque soit la méthode de projet choisie ; phases dont l'enchaînement et le temps d'exécution déterminent la durée totale de l'exécution du projet. Ce qui permet alors au chef de projet d'estimer le nombre de ressources nécessaires à chaque phase, et de voir comment moduler ce nombre pour gagner du temps.

La rédaction d'un plan de projet s'avère donc nécessaire afin de séquencer les phases en série ou en parallèle (totale ou partielle), les lier les unes aux autres grâce à des contraintes de départ ou de fin, et enfin pour déterminer le chemin critique.

J'ai donc créé une première version du plan de projet en février 2011, dans le logiciel Microsoft Project. Ceci nous a permis alors d'obtenir d'une part la date possible de livraison de l'application, et d'autre part le nombre de ressources nécessaires au projet, ceci avec des hypothèses de disponibilité de ressources discutées avec le chef de service.

II.3.1 Définition des grandes étapes

Le plan de projet a été découpé en une série de sept grandes phases, qui représentent à la fois un type d'actions à mener et une partie précise de l'avancée du projet.

Tableau 6 : Liste des phases du plan de projet

Phase	Description
Préparation	Définition du périmètre du projet, rédaction de l'analyse d'impact de la version, réception du cahier des charges, …
Design	Ecriture des documents de design, des schémas explicatifs
Développement	Développement du paquet d'installation et des artefacts Business Objects, migration et écriture des tests, …
Testing	Ecriture des protocoles d'installation, installation de l'environnement de test, tests informels et formels, rapport de test,…
Livraison	Ecriture du document description de l'application, mise à disposition des documents de design, livraison de l'application,
Formations	Formations aux équipes informatiques locales et aux utilisateurs
Déploiements	Installation sur les instances des sites, prise en compte des spécificités technique de chacun d'entre eux

Comme on le voit dans l'extrait du plan de projet ci-dessous, ces phases peuvent donc être consécutives (testing après développement) mais aussi être effectuées en parallèle, dans le cas où cela permet de gagner du temps (design en parallèle du développement).

Sur conseil du Global Custodian, il a été décidé d'appliquer ce principe de parallélisme pour certaines étapes, dans le but de gagner du temps lors des travaux de développement.

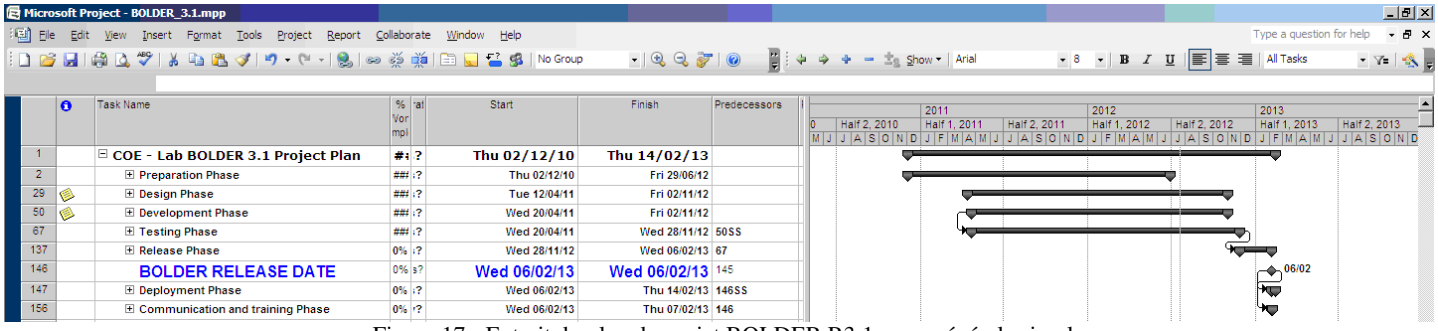


Figure 17 - Extrait du plan du projet BOLDER R3.1 : vue générale simple

Une période de quarante jours ('buffer') a été ajoutée après les travaux de développement et avant la date de livraison. Ce laps de temps a été mis en place suite aux incertitudes quant à la date de livraison, constatées au début du projet. Il servira à amortir d'éventuels retards, cumulés tout au long du projet.

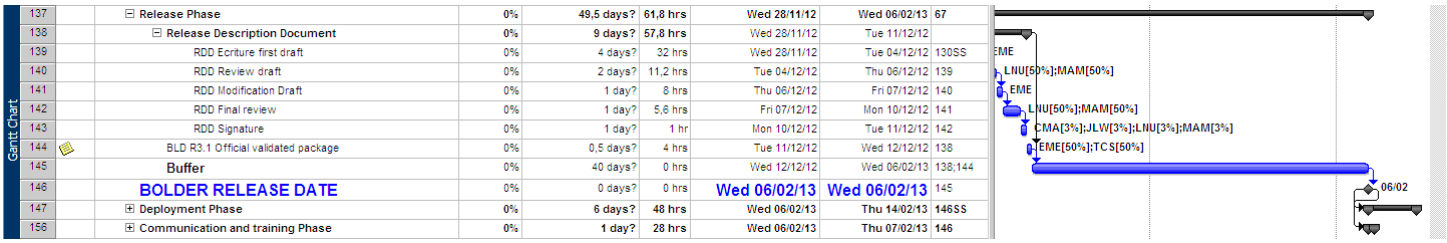


Figure 18 - Extrait du plan du projet BOLDER R3.1 : vue détaillée de réserve de temps

Le lecteur retrouvera en annexe un déroulé temporel de l'ensemble des actions du projet et un extrait plus complet du plan de projet, montrant le détail des phases les plus critiques[21].

[21] Voir annexe n° 4 'Déroulé temporel des actions du projet' et annexe n°5 : 'Extrait du plan de projet'

II.3.2 Importance relative de la date de livraison

Cependant, il doit être clair que cette estimation s'est faite dans un contexte particulier : d'une part, le cahier des charges livré par Lilly ne couvrait pas l'entièreté des besoins, ce qui mené à une sous-estimation des temps de développement, notamment de l'univers Business Objects.

D'autre part, le nombre de ressources estimées nécessaires n'a jamais été concrétisé au début du projet. Celui-ci a démarré avec une ressource exécutante et moi-même. D'autres personnes sont intervenues ponctuellement dans les premiers mois, mais cet état de fait ne m'a pas permis de mettre en place un plan de projet servant à mesurer la consommation de temps.

Enfin, il est à noter que du fait de la transformation de l'équipe Global MI&CS Lab, la date de livraison du projet en cours a perdu en importance, au profit des autres priorités. Ce qui s'est traduit par une allocation de ressources moindre, comme signalé, mais aussi une disponibilité réduite des experts technique et métier.

De mon côté, j'ai dû intégrer cette absence de contrainte, non pas en la considérant comme définitivement annulée, mais en continuant à faire avancer le projet du mieux possible dans le cas où la date de livraison redeviendrait critique.

Afin de réduire le risque d'un ralentissement du projet, j'ai assuré une collaboration continue avec les deux nouvelles ressources intégrées en septembre 2011, pour pouvoir les faire rapidement monter en compétence et leur attribuer les tâches les plus complexes et les plus stimulantes. Le but était de profiter au maximum de leur présence limitée dans le temps, et de leur expertise, afin d'obtenir des développements rapides et bonne qualité. Le fait de travailler en commun en continu a permis de tisser des relations de travail bien plus efficientes que celles que j'ai pu créer avec l'équipe offshore.

Cette volonté de garder ces deux ressources occupées continuellement a également été une aide pour éviter leur démotivation, face aux incertitudes des contraintes de temps ou face

à des demandes fonctionnelles délivrées au fur et à mesure des points avec l'architecte technique.

II.3.3 Difficiles estimations de temps

La première version du plan de projet a ainsi reflété les obstacles rencontrés lors de l'évaluation de la durée de chaque tâche. Réalisée en commun avec le Global Custodian, elle a été rendue difficile par plusieurs paramètres :

- comme indiqué, le nombre de ressources a été théorique et fluctuant dès le début du projet, avec des niveaux technique différents et ne m'a pas permis d'établir clairement le nombre de jours nécessaires pour chaque tâche.

- le cahier des charges ne mentionne que les ajouts de champs de la base de données, en omettant l'aspect fonctionnel de l'application, à savoir les artefacts Business Objects. L'estimation de temps du développement de l'univers s'est faite uniquement sur l'ajout des nouveaux champs, sans envisager d'autres développements nécessaires.

- enfin, le projet BOLDER R3.1 a impliqué des activités pour lesquelles ni l'équipe Lab ni moi-même ne connaissions de précédent. Les développements des fichiers Ada!Rec, ou plus tard l'écriture des tests dans Quality Center, ont été difficiles à estimer, en raison de la spécificité de chaque fichier de code ou de chaque de cas de test. De plus, les travaux de développement des premiers mois ont été effectués par l'équipe d'analystes offshore. Il a été difficile d'obtenir des hypothèses de productivité des ressources offshore, au vu des résultats constatés dans le cadre de ma mission de support.

La première estimation de temps de réalisation a donc été basée sur des hypothèses découlant de l'analyse des contraintes de la situation. Elle a été acceptée comme telle par le client Lilly, représenté par le Global Custodian. Afin de pouvoir obtenir une date de livraison théorique avant 2013, il m'a été demandé d'estimer un temps de réalisation modulé par les différents nombres de ressources.

Basé sur ces estimations, le chef de service de l'équipe Global MI&CS Lab a pu faire des demandes de ressources supplémentaires, en présentant la situation existante et les

accélérations potentielles en cas de nouvelles ressources ; ressources qui ont donc été intégrées fin septembre 2011.

Cependant l'incertitude quant à la date de livraison a perdurée, suite d'une part à la focalisation d'une partie des ressources sur la transition vers le nouveau prestataire offshore, et d'autre part à la réévaluation des travaux nécessaires sur l'univers Business Objects.

Le plan de projet a donc surtout servi dans une grande partie du projet de fil directeur dans les tâches à faire effectuer, en fonction des disponibilités des différentes ressources. Il a aussi été maintenu à jour, en intégrant à chaque fois les nouvelles dates de tâches au fur et à mesure de leurs avancées, mais aussi de l'évolution des hypothèses liées aux contraintes extérieures.

Il est à noter que j'ai, à chaque réunion de projet, alerté le Global Custodian sur cette échéance incertaine. La phase de transformation de l'équipe Global MI&CS Lab étant prioritaire, ce retard persistant a été accepté comme tel.

Cette situation a évoluée à la fin de la phase de transition vers le nouveau prestataire (juin 2012), où la date de livraison est redevenue contraignante.

II.3.4 Renouvellement de la contrainte temps

C'est donc à la fin du mois de juin 2012 que le plan de projet a à nouveau joué pleinement son rôle d'indicateur d'avancement.

Lors d'un état des lieux du projet, j'ai signalé qu'au vu des développements encore en cours, et du reste des étapes à accomplir, le projet ne saurait se terminer avant février 2013.

Cette échéance trop tardive a amené le Global Custodian a demandé quelles actions pouvaient mises en place pour rapprocher la date de fin de projet.

J'ai donc réévalué l'estimation de temps des phases restantes, notamment celle de la phase de testing informel.

Afin de faire face à ce nouveau besoin, j'ai travaillé sur deux axes d'amélioration potentielle, qui seront détaillés dans la partie V-1 'Accélération du projet' : l'augmentation des ressources et le redécoupage de l'enchainement des tests.

Je ne m'étendrai donc pas plus sur cette action dans cette partie, mais il est à noter qu'à partir du moment où la contrainte de temps a été concrétisée, la mise en place d'actions pour l'atteindre a été plus facile.

J'ai en effet pu estimer en fonction du temps de travail dédié au testing en quelle mesure le nombre de ressources obtenu pouvait accélérer le projet. Cela m'a permis aussi de travailler plus en finesse sur le plan de test.

II.4 Chapitre quatre : les premières étapes techniques

II.4.1 Les tests de compatibilité Ada!Rec vs Oracle 11

II.4.1.1 Le contexte

Dans le cadre du cycle de vie des applications et des éléments d'infrastructures informatiques, une action de migration des bases Oracle vers la version 11 a été engagée au courant de l'année 2010.

En ce qui concerne les bases BOLDER, la première étape a consisté à vérifier la compatibilité de l'outil de transfert Ada!Rec avec une base BOLDER migrée en version 11.

Ces travaux ont eu lieu au courant de l'automne 2010, en amont du projet BOLDER R3.1.

Les tests de la compatibilité de cet outil avec les bases Oracle source ou destination ont une importance particulière. Des résultats négatifs conditionnent en effet soit le changement d'outil ETL, soit la non-migration de la base Oracle concernée.

II.4.1.2 Les tests effectués

Comme indiqué en première partie[22] Ada!Rec extrait d'EuroLIMS des données de manière incrémentale (transfert journalier) ou complète (installation, réplication ou restauration du système), dans des fichiers plats.

Les données sont ensuite chargées dans la base de données cible BOLDER à partir de ces mêmes fichiers. Les tests ont donc portés sur cette partie du processus.

Une copie d'une base de données BOLDER d'une instance globale de test a été effectuée par un administrateur de bases de données, vers la version Oracle 11.

La base de données a été copiée intégralement, avec les données et le schéma du compte applicatif, ainsi que les comptes applicatifs secondaires.

[22] Première partie, chapitre deux, section quatre 'L'outil de reporting BOLDER'.

J'ai ensuite paramétré l'outil de transfert pour qu'il pointe vers cette nouvelle base et ait lancé un transfert complet depuis la base EuroLIMS cible.

Le transfert s'est passé sans incident et l'outil de comparaison du nombre d'enregistrements extraits et intégrés n'a présenté aucune erreur. J'ai ensuite réalisé un transfert incrémental, en créant et modifiant des données dans la base EuroLIMS, puis en lançant l'outil de transfert. A nouveau, l'analyse des journaux de transfert n'ont montré aucune erreur ni anomalie.

Ceci a donc permis la décision de continuer à utiliser l'outil Ada!Rec pour le transfert de données vers BOLDER, et à entamer la migration des bases BOLDER vers Oracle 11. Cette migration a été intégrée dans le projet BOLDER 3.1 pour les bases de production.

II.4.2 Migration de la plate-forme Business Objects

Une des autres composantes du projet BOLDER R3.1 est le passage de Business Objects XI r2 vers la version XI r3.

J'ai donc demandé en février 2011, à l'équipe en charge de cette plate-forme, la création de la structure Business Objects XI r3, pour l'environnement de développement, et la migration de l'univers et des rapports globaux validés, issus de la version BOLDER R3.0.

Cette action s'est inscrite dans le cadre plus large de la migration de toutes les applications basées sur Business Objects chez Lilly, au niveau mondial.

II.4.3 La formation à Business Objects XI r3

Afin de me préparer aux travaux sur Business Objects, j'ai suivi une formation en ligne financée par mon entreprise. Cette formation, dispensée par l'entreprise SAP, présente de manière très détaillée les différentes fonctionnalités de l'outil de modélisation de rapports Business Objects (Report Designer).

J'ai choisi à dessein ce mode de formation, suite à des expériences positives avec les cours en ligne du CNAM.

Il présente des avantages conséquents par rapport à des cours en présentiel :

- la possibilité d'étaler dans le temps les sessions de formation, au lieu de les regrouper sur quelques jours. Le module de formation est ouvert sur six mois et m'a permis de le suivre en fonction de l'avancement du projet et de la réalisation des travaux sur l'univers Business Objects.
- la mise à disposition de supports électroniques, ce qui permet de pouvoir les consulter rapidement sur tout ordinateur lié au réseau Internet, sans devoir s'encombrer de support papier.
- l'économie du temps de transport et la possibilité de suivre la formation hors temps de travail, ce qui permet de ne pas impacter la prestation en cours.

II.4.4 L'écriture du Release Impact Assessment

Ce document pourrait s'appeler en français 'Analyse d'impact et plan de validation de la nouvelle application'. Il est approuvé par le Global Custodian et le Global Owner[23] de l'application, pour l'ensemble des modifications apportées à la nouvelle application.

J'ai rédigé la première version de ce document au printemps 2011. J'y ai décrit les impacts impliqués par le développement, la validation et la livraison d'une nouvelle version de la version BOLDER R3.1, ceci à la fois pour les systèmes associés (EuroLIMS, l'outil Ada!Rec), pour l'équipe de développement et pour ses futurs utilisateurs.

Une présentation détaillée des modifications techniques y est présenté, comprenant toutes les demandes de modifications qui ont été retenues pour cette version. Ces demandes sont soient issues d'investigations concernant des incidents informatiques, soit des demandes d'amélioration venant des sites ou de l'équipe Global MI&CS Lab.

Un état des lieux de la documentation de validation a été également faite, incluant les documents à créer ou à modifier. Enfin, une première estimation de temps a été faite, qui ne fut pas considérée comme définitive, mais permit d'envisager l'impact en terme de charge de travail (jours/homme).

Enfin, l'impact pour les utilisateurs a est également été abordé, pour ce qui concerne la partie formation.

[23] Global Owner : voir glossaire.

II.4.5 La modélisation de la base de données

Les travaux de modélisation de la base de données BOLDER ont pu alors commencer, au mois d'avril 2011. Ils ont duré un mois, au terme duquel les premiers développements ont démarré, avec les ressources de l'équipe offshore.

Le nombre de tables EuroLIMS transférées dans BOLDER (32) a été plus que doublé : les tables devant soit être transférées quasiment à l'identique, soit être fusionnées avec d'autres, de manière à dénormaliser leur schéma en flocon de la base de données EuroLIMS.

Il a été décidé de simplifier la modélisation de l'univers afin notamment d'optimiser les temps de réponses des rapports. De plus, les utilisateurs n'ont ou n'auront pas connaissance de toutes les relations possibles entre les tables BOLDER. Le choix a donc été de fusionner le plus possible les tables de dimension avec les tables de faits.

Prenons comme exemple la table TEST. Dans EuroLIMS, elle contient des données propres aux tests, mais aussi des champs permettant des jointures avec d'autres tables.

On retrouve ainsi un modèle en étoile, qui permet à l'application de gérer de manière efficace la sélection et la mise à jour des données.

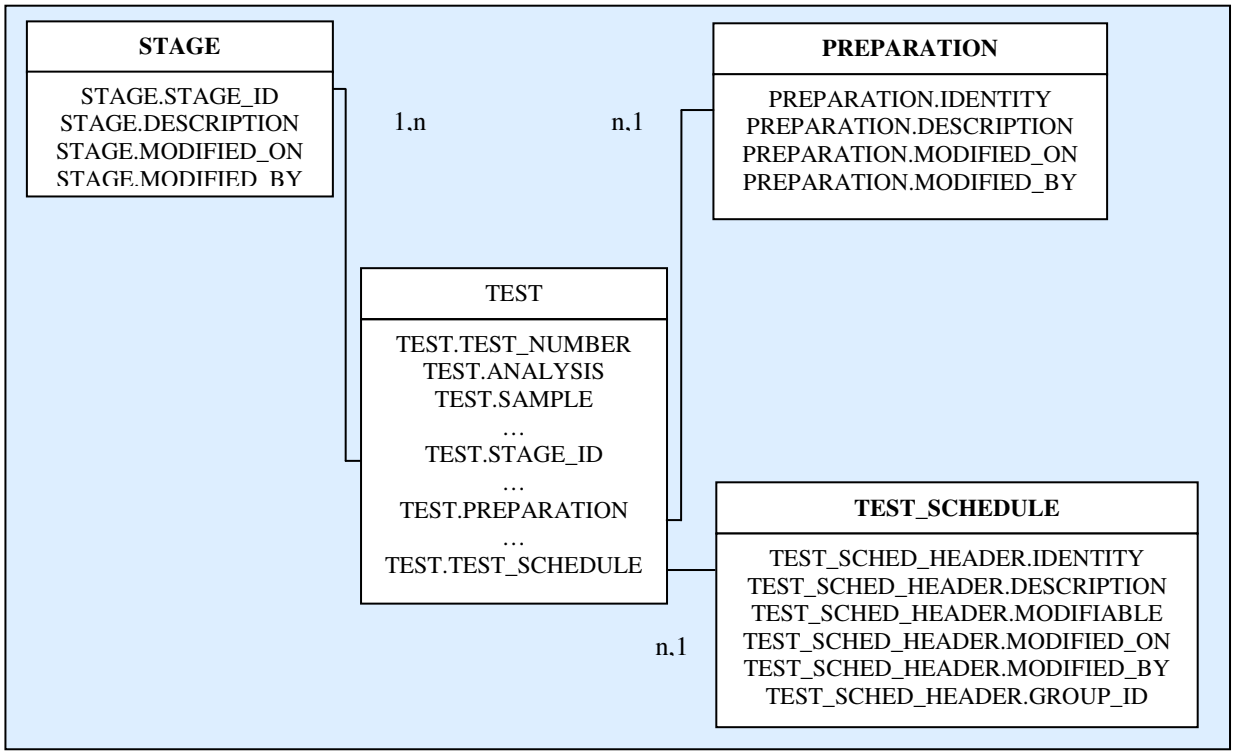


Figure 19 - Modèle en étoile pour la table TEST dans EuroLIMS.

Le travail de modélisation a consisté à dénormaliser ces tables EuroLIMS pour fusionner un certain nombre de leurs champs dans la table TEST de BOLDER.

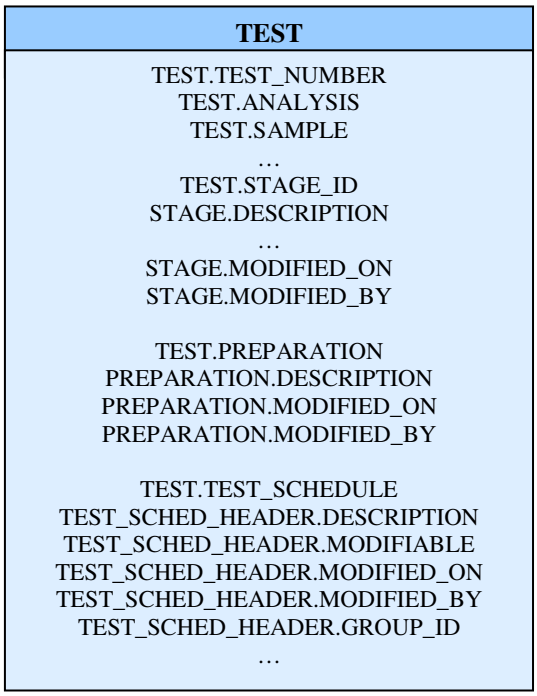


Figure 20 - Schéma du modèle dénormalisé pour la table TEST dans BOLDER

Les données des différentes tables concernant un même enregistrement de la table TEST d'EuroLIMS ont été rassemblées sur une même ligne dans la table BOLDER, table qui contient forcément des données redondantes et qui se trouve être plus large en taille que sa table source EuroLIMS[24].

Ainsi, les utilisateurs de BOLDER R3.1, qui ne disposeront pas forcément de la connaissance de l'application EuroLIMS pour faire le lien entre un test et la donnée de 'stage', auront à disposition dans BOLDER R3.1 ces informations disponibles dans la même table.

[24] Voir annexe n°6 : Extrait du document de design de la base de données : table TEST

III Troisième partie : migration et écriture des tests

[Octobre 2011 - Mars 2012]

III.1 Chapitre un : migration des tests vers une solution électronique

III.1.1 Situation préexistante

Une partie de la documentation de validation, sous format papier, est constituée de l'ensemble des cas de tests créés pour valider chaque nouvelle version d'EuroLIMS ou de BOLDER. Il peut s'agir de tests système, d'intégration ou de sécurité, permettant de valider que chaque expression de besoins listée dans le cahier des charges a été satisfaite. D'autres tests peuvent se rajouter, suite à des corrections de bogues ou des évolutions techniques non demandées par les utilisateurs.

III.1.2 Les raisons de la migration

Comme il a été indiqué, EuroLIMS est une application placée en statut « gelé », décision prise par le comité de gouvernance des applications laborantines de l'équipe Global MI&CS Lab. Ce qui implique qu'aucune nouvelle version n'a été réalisée depuis cette décision, mise à part quelques cas particuliers.

Il n'y a donc pas eu d'effort de migration des tests vers le format électronique, du fait du statut figé d'EuroLIMS et du ratio coût de migration versus retour sur effort.

BOLDER ayant peu évolué (cinq versions en dix ans) et été considéré comme une application auxiliaire d'EuroLIMS, l'ensemble de ses tests de validation est également resté au format papier.

Ainsi, au commencement du développement de la version BOLDER R3.1, les tests d'intégration ou les tests système de BOLDER à exécuter étaient imprimés depuis leur modèle électronique, exécutés, puis documentés manuellement.

Le cahier des charges de la nouvelle version de BOLDER requiert le transfert de plus de trente nouvelles tables EuroLIMS, de manière simple ou combinée. Ceci implique qu'un grand nombre de cas de tests sera à ajouter au corpus déjà existant.

En ce qui concerne le projet BOLDER R3.1, le testing était dès le démarrage prévu d'être exécuté avec les ressources offshore. Cependant, il a été constaté, lors des activités de support, que l'envoi et la revue de protocoles au format papier peut être complexe à gérer (scan, envoi par mail, impression,…). Faire exécuter et de revoir de cette manière plus de deux cents tests ralentirait considérablement les phases de tests.

Le projet BOLDER R3.1 a donc inclus en sus des actions de développement, une migration des cas de tests existants dans l'application informatique Quality Center, de l'éditeur Hewlett Packard, dont les services informatiques de Lilly se sont dotés en 2007 afin de s'affranchir de la gestion et de l'archivage des documents de test en papier.

III.1.3 La migration des cas de tests BOLDER

J'ai initié cette migration en octobre 2011 par un état des lieux de l'existant et ai choisi quels tests devaient être migrés dans l'application Quality Center, en tenant compte de la pertinence des tests pour la version BOLDER R3.1 et en ne migrant pas ceux créés pour des points techniques devenus obsolètes.

J'ai choisi de confier cette tâche aux deux nouvelles ressources basées à Fegersheim pour des raisons de simplicité et proximité.

Simplicité, car arrivant sur le projet, elles étaient en début de formation et leurs connaissances techniques et fonctionnelles de BOLDER à ce moment précis étaient peu élevées. L'utilisation de Quality Center est relativement simple, il était donc aisé pour elles de migrer les tests BOLDER tout en acquérant des connaissances sur le contexte BOLDER.

Proximité, parce que la migration des tests a impliqué une refonte du système d'identification des cas de tests, mais aussi une restructuration des instructions de tests. De

nombreux échanges et réflexions communes ont eu lieu entre les deux ressources et moi-même et celles-ci auraient pris bien plus de temps si ces tâches avaient été dédiées à des ressources offshore.

III.2 Chapitre deux : (re)écriture des nouveaux cas de tests

III.2.1 Refonte du système d'identification des tests

Les cas de tests BOLDER sont nommés selon un schéma récurrent. Nous trouvons ci-dessous un exemple simple :

ITS.SEC.BLD.2.0

qui représente l'ensemble de cas de test d'intégration (Integration TeSt), pour le module sécurité (SECurity) pour l'application BOLDER (BoLDer). Le numéro 2.0 indique qu'au moins un autre ensemble de cas de tests, pour le même module, existe ou a existé en parallèle de celui-ci (par exemple : ITS.SEC.BLD.1.0). Une version existe pour chaque ensemble et est renseignée dans le document en lui-même.

La problématique de la migration fut de conserver ce système d'identification, pour permettre de retrouver la filiation entre les tests papier et leurs successeurs, dans le système Quality Center.

III.2.2 Réécriture des tests

Le format d'écriture des étapes de test dans Quality Center a imposé de revoir la structure interne de ceux-ci.

A l'origine, les tests papier présentaient des scénarios de tests ('Tests scripts') comprenant des cas de tests ('Tests cases').

Leur dénomination reprenait une structure bien précise, liant les requis aux cas de tests. Ainsi, les requis RQ.TR.17.x étaient testés dans le script de test ITS.TRN.17, qui comprenaient les cas IC.TRN.17.x, testant le transfert de données suite à insertion, modification ou suppression des données dans EuroLIMS.

Cette hiérarchie à deux niveaux permettait donc d'exécuter des cas de tests en se basant sur les mêmes données d'essai, et de tester dans le même script un ensemble commun de requis issus du cahier des charges.

Le passage à l'outil informatique a imposé une nouvelle structure aux tests. En effet, Quality Center ne permet pas la même hiérarchisation 'scénarios de tests' - 'cas de tests' : un seul niveau de classement est possible. Le résultat en est que les scénarios de tests durent être déstructurés et répartis en jeux de tests séparés.

Dans la mesure où les anciens tests de transfert de données se succédaient et partageaient des données d'essais communs, ces jeux de tests ont été réécrits et regroupés de manière à prévoir une exécution liée : les tests de modification de données ne peuvent être exécutés tant que les tests d'insertion de données n'ont pas été complétés. Ce qui permet de réutiliser un jeu de données communes.

Dans l'exemple ci-dessous, l'ensemble des nouveaux tests IT.TRN.17.x vérifient le transfert correct de données d'EuroLIMS pour chaque cas de figure possible (par exemple : insertion, ou modification).

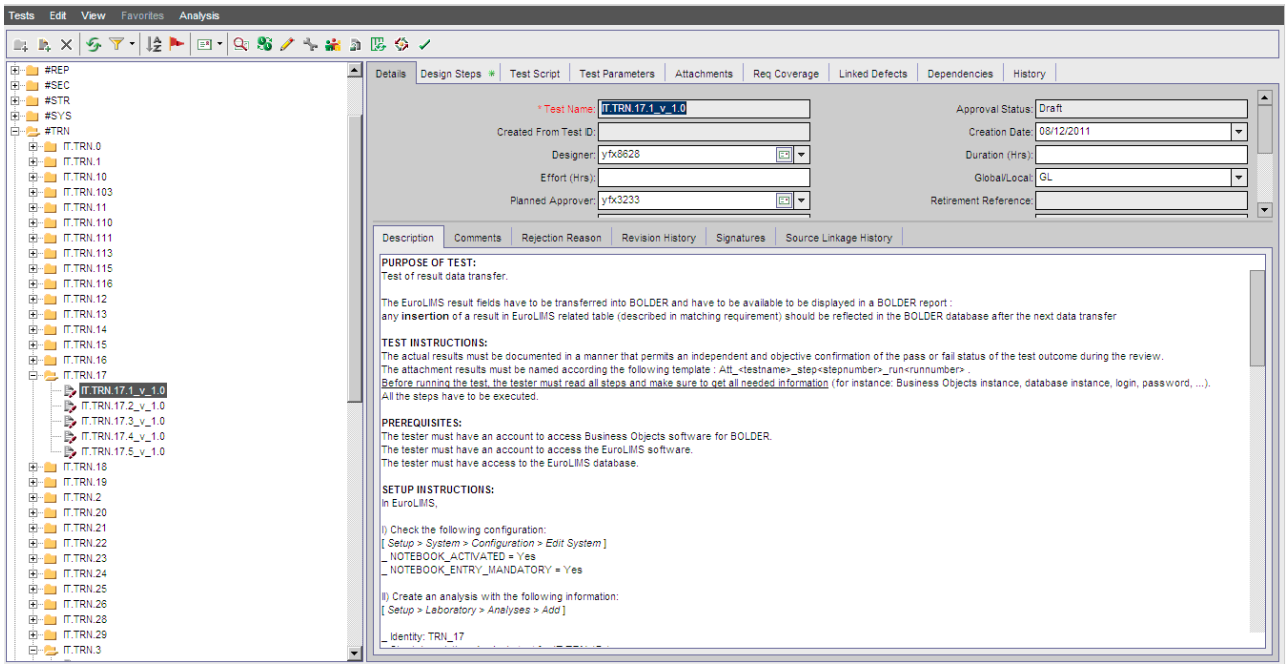


Figure 21 - Extrait de la série de tests BOLDER dans Quality Center : tests IT.TR N17.x

Les instructions d'exécution de chaque test ont été réécrites, en prenant soin de les détailler plus précisément que dans leur version papier. En effet, j'ai jugé que les futurs testeurs potentiels ne seraient pas forcément des ressources expérimentées dans l'utilisation d'EuroLIMS et de BOLDER, de Quality Center, ni dans la documentation des tests exécutés.

Afin d'éviter toute perte de temps et minimiser les erreurs d'interprétation des instructions lors de l'exécution du testing par des exécutants novices, j'ai demandé aux rédacteurs des tests d'être le plus précis possible dans l'écriture des instructions de mis en place, en indiquant par exemple le chemin complet à utiliser dans les menu applicatifs.

Bien que cette réécriture ait eu un surcoût en temps non négligeable par rapport à une migration par copier-coller, elle a été faite dans le but de gagner du temps lors de l'exécution des tests, en cadrant très fortement les testeurs, non connus à l'époque de la migration des tests.

III.2.3 Mise à jour de la documentation relative aux tests

Le format électronique différant des documents papier, il m'a fallu prévoir une refonte des procédures d'écriture des tests, et de la documentation de ceux-ci lors de leur exécution. Ce document ('BOLDER Overall Test Plan') a été intégré au cursus de formation des ressources travaillant sur le projet BOLDER R.3.1, comme pré-requis au travail en environnement validé.

Par ailleurs, il existe un document maître servant à faire le lien entre le cahier des charges et les tests.

Il s'agit de la matrice de traçabilité, qui reprend l'ensemble des tests, qui exécutés sans échec, permettent de prouver qu'un requis issu du cahier des charges a été satisfaite.

Ce document est donc capital pour la phase de validation car il permet la mise en place de la stratégie de test, et est à mettre à jour pour tout ajout ou retrait de cas de tests.

C71 — Any insertion of a stability protocol in the EuroLims should be reflected in the Bolder database after the next data transfer

	A	B	C	D	E
1	**Test Quality Center**	**Requirement Number BOLDER 3.1**	Description BOLDER 3.1	Design ID	Test Level
71	**IT.TRN.12.1**	**RQ.TRN.12.2**	Any insertion of a stability protocol in the EuroLims should be reflected in the Bolder database after the next data transfer	BLD-VAL-Technical Informations on Bolder Data Transfer Process	I
72	**IT.TRN.12.2**	**RQ.TRN.12.3**	Any modification of a stability protocol in the EuroLims should be reflected in the Bolder database after the next data transfer	BLD-VAL-Technical Informations on Bolder Data Transfer Process	I
73	**Product Lots Transfer**				
	IT.TRN.13.1	**RQ.TRN.74**	The EuroLIMS fields that have to be transferred into BOLDER and to be available to be displayed in a Bolder report for EuroLIMS retirement purpose are: PRODUCT_LOT.CREATED_BY PRODUCT_LOT.CREATED_ON PRODUCT_LOT.DESCRIPTION PRODUCT_LOT.EXPIRY_DATE PRODUCT_LOT.INSPECTION_LOT_NO PRODUCT_LOT.ITEM_CODE_ID PRODUCT_LOT.JOB_TEMPLATE PRODUCT_LOT.LOT_ID PRODUCT_LOT.MANUFACTURE_DATE PRODUCT_LOT.NEXT_INSPECTION_DATE PRODUCT_LOT.SAMPLE_TEMPLATE PRODUCT_LOT.SAP_BATCH PRODUCT_LOT.SAP_MATERIAL PRODUCT_LOT.SOURCE_ID	BDR-VAL-Bolder Database TI	I

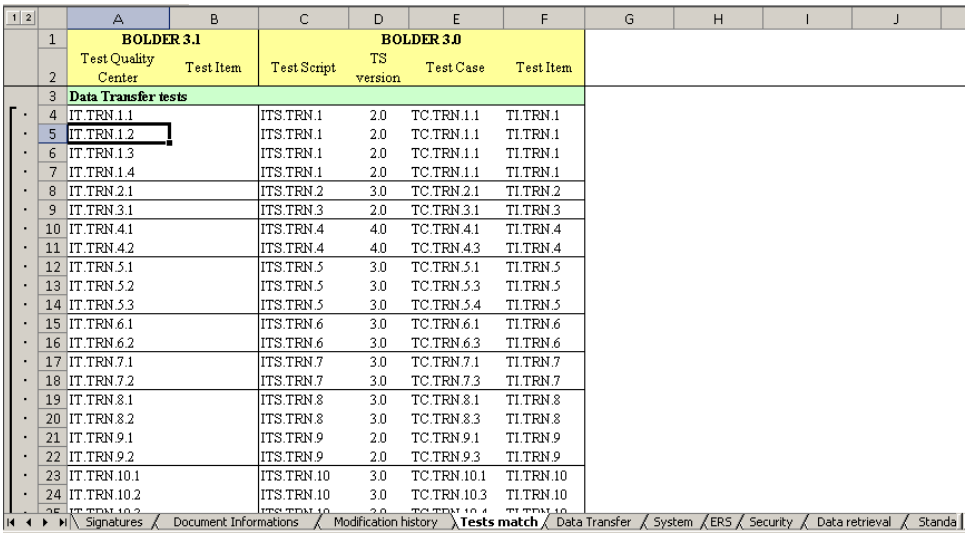
Figure 22 - Extrait de la matrice de traçabilité, pour le transfert de la table ‘Product lot’.

Nous retrouvons dans l’extrait ci-dessus une demande du cahier des charges : il s’agit de la demande de transfert d’un certain nombre de champs de la table ‘product_lot’, identifiée par ‘RQ.TRN.74’, soit ‘REQuirement.TRaNsfer, numéro 74’.

On le voit ci-dessus, à chaque requis correspond un jeu de test : nous avons ici le cas de test IT.TRN.13.1. Il s’agit du test d’intégration, pour le transfert de données, numéro 13.1. Il reprend la numérotation du requis de la version précédente de BOLDER, RQ.TRN.13, remplacé par le RQ.TRN.74.

J’ai décidé de garder lors de la migration des tests leur nom d’origine, afin de permettre dans un futur projet de pouvoir faire le lien entre les différentes versions des tests.

Comme il est critique d’un point de vue validation de conserver la progression entre les différentes versions de tests, j’ai demandé à ce que soit créé dans la matrice de traçabilité un onglet explicitant la correspondance entre les anciens tests papier et leur correspondant électronique.



	BOLDER 3.1		BOLDER 3.0			
	Test Quality Center	Test Item	Test Script	TS version	Test Case	Test Item
3	**Data Transfer tests**					
4	IT.TRN.1.1		ITS.TRN.1	2.0	TC.TRN.1.1	TI.TRN.1
5	IT.TRN.1.2		ITS.TRN.1	2.0	TC.TRN.1.1	TI.TRN.1
6	IT.TRN.1.3		ITS.TRN.1	2.0	TC.TRN.1.1	TI.TRN.1
7	IT.TRN.1.4		ITS.TRN.1	2.0	TC.TRN.1.1	TI.TRN.1
8	IT.TRN.2.1		ITS.TRN.2	3.0	TC.TRN.2.1	TI.TRN.2
9	IT.TRN.3.1		ITS.TRN.3	2.0	TC.TRN.3.1	TI.TRN.3
10	IT.TRN.4.1		ITS.TRN.4	4.0	TC.TRN.4.1	TI.TRN.4
11	IT.TRN.4.2		ITS.TRN.4	4.0	TC.TRN.4.3	TI.TRN.4
12	IT.TRN.5.1		ITS.TRN.5	3.0	TC.TRN.5.1	TI.TRN.5
13	IT.TRN.5.2		ITS.TRN.5	3.0	TC.TRN.5.3	TI.TRN.5
14	IT.TRN.5.3		ITS.TRN.5	3.0	TC.TRN.5.4	TI.TRN.5
15	IT.TRN.6.1		ITS.TRN.6	3.0	TC.TRN.6.1	TI.TRN.6
16	IT.TRN.6.2		ITS.TRN.6	3.0	TC.TRN.6.3	TI.TRN.6
17	IT.TRN.7.1		ITS.TRN.7	3.0	TC.TRN.7.1	TI.TRN.7
18	IT.TRN.7.2		ITS.TRN.7	3.0	TC.TRN.7.3	TI.TRN.7
19	IT.TRN.8.1		ITS.TRN.8	3.0	TC.TRN.8.1	TI.TRN.8
20	IT.TRN.8.2		ITS.TRN.8	3.0	TC.TRN.8.3	TI.TRN.8
21	IT.TRN.9.1		ITS.TRN.9	2.0	TC.TRN.9.1	TI.TRN.9
22	IT.TRN.9.2		ITS.TRN.9	2.0	TC.TRN.9.3	TI.TRN.9
23	IT.TRN.10.1		ITS.TRN.10	3.0	TC.TRN.10.1	TI.TRN.10
24	IT.TRN.10.2		ITS.TRN.10	3.0	TC.TRN.10.3	TI.TRN.10

Signatures / Document Informations / Modification history / **Tests match** / Data Transfer / System / ERS / Security / Data retrieval / Standa

Figure 23 - Extrait de la matrice de traçabilité, onglet montrant la correspondance entre anciens et nouveaux tests.

Sur ce modèle, la matrice de traçabilité a donc été mise à jour en intégrant les nouveaux requis du cahier des charges et en incluant pour chacun d'entre eux les tests à effectuer, qu'il s'agisse d'un test migré ou d'un test nouvellement créé.

IV Quatrième partie : développement de l'application BOLDER

[Novembre 2011 - Juillet 2012]

L'application BOLDER R3.0 est composée de trois composants techniques ; l'outil d'interface, la base de données et l'outil de reporting pour permettre une recherche intuitive pour des utilisateurs non-informaticiens.

Je vais décrire dans les pages manière dont j'ai dirigé ces travaux des développements de ces composants, les méthodes de travail que j'ai mises en places et la contribution que j'y ai apportée.

IV.1 Chapitre un : organisation des activités

IV.1.1 La problématique du suivi des fichiers

BOLDER R3.1 compte soixante tables, qui intègrent les données de soixante-dix tables EuroLIMS, grâce à deux modes de transferts différents (complet et incrémental). Chaque interface est composée de deux fichiers Ada!Rec : un pour l'extraction des données, un pour l'insertion.

Rajoutons les scripts de création des tables d'interfaces côté EuroLIMS, les scripts de création de la base de données BOLDER et enfin les scripts servant au fonctionnement de l'interface. Le paquet d'installation de BOLDER R3.1 contient donc près de trois cent soixante fichiers, dont une bonne partie a été ajoutée ou modifiée dans le paquet d'installation de la version R3.0, comme on peut le voir dans le tableau ci-dessous.

Tableau 7 : Récapitulatif des fichiers des versions BOLDER R3.0 et R3.1

Type de fichier	BOLDER 3.0		BOLDER 3.1	Nouveaux fichiers	Existants modifiés	Existants non-modifiés
Ada!Rec	148		304	156	118	30
SQL	27		41	14	23	4
Unix	9		13	4	5	4
Total	**184**		**358**	**174**	**146**	**38**

Cela implique donc de pouvoir suivre le développement de chaque fichier, de pouvoir tracer à tout moment où il se trouve, et qui l'a modifié en dernier.

Il est aussi nécessaire de garantir son unicité de modification à un instant donné et que la nouvelle version du fichier soit réintégrée dans le paquet d'installation de manière adéquate.

IV.1.1.1 Besoin d'une solution de suivi simple et commune

Il m'a donc fallu trouver une solution pour assurer ce suivi, solution qui soit comprise et admise par l'ensemble des personnes intervenant sur le développement.

Avant de nous pencher sur les solutions logicielles (payantes ou libres) existant sur le marché ou chez Lilly, je vais, dans le schéma ci-dessous, m'efforcer d'analyser en détail les processus participant au cycle de vie d'un fichier de code.

Cette démarche va nous permettre d'identifier les acteurs du processus et surtout les étapes clés qui permettent de garantir le suivi et l'unicité des fichiers de code.

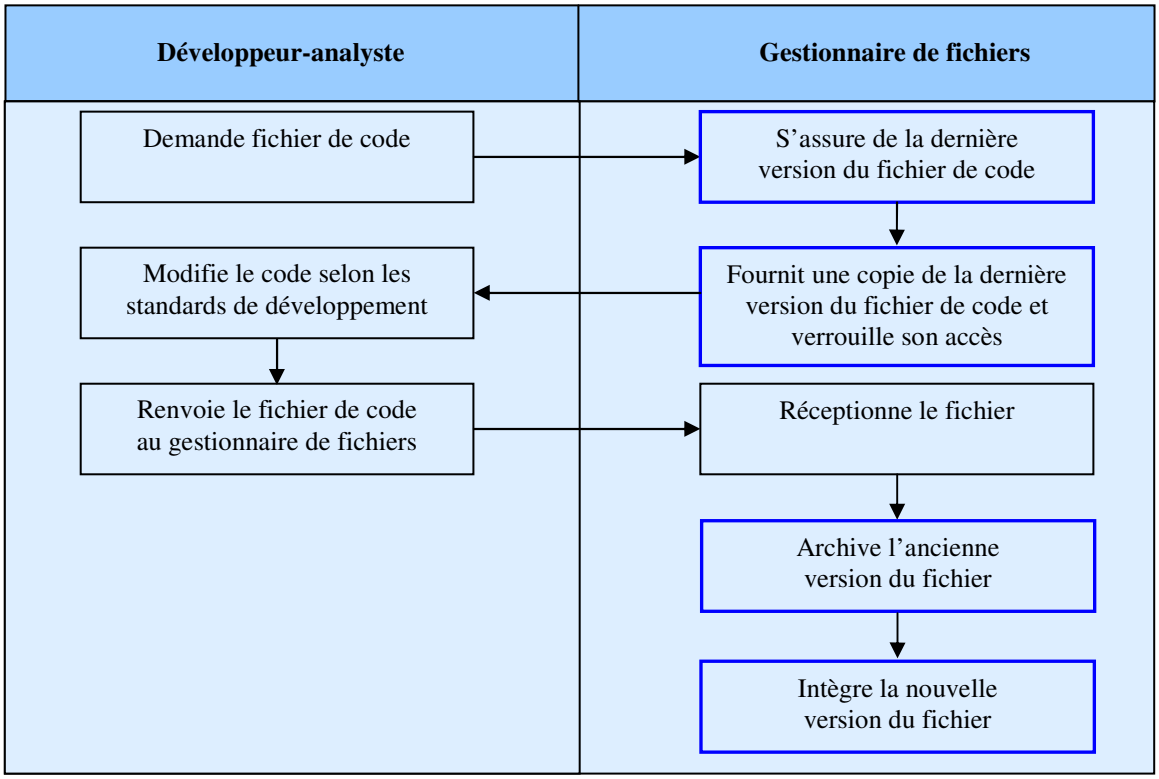


Figure 24 - Processus de base de gestion des fichiers de code :
les rectangles bleus représentent des points de contrôle

Comme on le voit dans ce schéma, le développeur-analyste ne fait que recevoir et renvoyer les fichiers modifiés. La responsabilité du suivi des fichiers et de leur intégration dépend donc exclusivement du gestionnaire de fichier, que cette fonction soit assurée par une seule entité, que ce soit une personne ou un logiciel.

Cependant, la solution utilisée doit pouvoir être comprise et utilisable par tous les acteurs, afin de ne pas pénaliser les travaux de développements. Par exemple, une application de suivi de fichiers trop complexe pourrait devenir un frein.

Il m'a donc fallu faire un état des lieux des solutions existantes et comprendre laquelle pourrait être la plus adaptée.

IV.1.1.2 Le processus au sein de l'équipe Global MI&CS Lab

L'équipe MICS Lab produit des développements informatiques depuis de nombreuses années, et selon les situations, différentes méthodes de gestion de suivi des fichiers de code ont été mises en place.

Le point central de la gestion du suivi des fichiers est alors assuré par le(s) coordinateurs du support EuroLIMS/BOLDER basés à Fegersheim. Celle-ci consiste à délivrer les fichiers de code à aux analystes offshore, puis à réceptionner les fichiers et assurer leur revue et leur intégration dans le paquet d'installation.

Les fichiers sont stockés dans un répertoire de fichiers inaccessible pour les analystes offshore. Ils sont de plus sauvegardés tous les jours par un système automatisé.

Ce processus repose sur la discipline et la vigilance de ceux qui le mettent en œuvre : une erreur de manipulation peut en effet entrainer la perte de fichiers de code validés et officiels.

IV.1.1.3 La solution SharePoint

SharePoint est déployé chez Lilly de manière intensive. Chaque équipe ou même chaque personne est susceptible d'utiliser un ou plusieurs sites collaboratif(s). C'est donc un outil de travail quotidien, d'utilisation relativement simple, et qui est supporté de manière quotidienne par les équipes informatiques idoines.

Comme on le sait, l'application collaborative Microsoft SharePoint permet de stocker des fichiers, de gérer leur accès, leur version et leur statut d'entrée/sortie ('check-in/check-out').

L'équipe de support EuroLIMS/BOLDER l'a utilisé pendant un certain temps pour ses diverses activités, pour l'échange de fichiers, et c'est lors d'une phase de développement que nous nous sommes rendu compte que l'anti-virus associé à SharePoint peut considérer certains fichiers de code comme des virus potentiels et les supprimer.

La définition des virus potentiels d'un anti-virus évoluant avec le temps, c'est donc un risque pour toute équipe de développement d'utiliser cet outil comme gestionnaire de fichiers de code.

D'autre part, le système de gestion des fichiers de Sharepoint ne permet pas d'assurer une unicité dans l'utilisation des fichiers : les accès concurrentiels sont autorisés sur ceux-ci, malgré certains garde-fous en place. Le respect de ceux-ci dépend uniquement de la vigilance et de leur compréhension par les utilisateurs, sous peine de voir un fichier modifié par deux personnes au même moment ou bien incorrectement replacé dans le site collaboratif.

Le paquet d'installation BOLDER R3.1 comporte au final quelques trois cent soixante fichiers. Ceci multiplie d'autant les risques d'erreur, qui peuvent avoir des effets découverts bien plus tard.

Au vu de ces constatations, j'en ai déduit que l'outil collaboratif SharePoint n'était pas indiqué car il ne garantit pas que les fichiers du paquet d'installation puissent être gérés de manière automatique pour garantir leur unicité et leur pérennité tout au long du cycle de développement.

IV.1.1.4 Les logiciels du marché

Les applications permettant la gestion des fichiers de code et leur version sont dénommés 'Logiciels de Gestion de Versions' (LGS) ou 'Version Control System' (VCS). De nombreux produits disponibles sur le marché nous indiquent que cette problématique auquel j'ai été confronté est prégnante et que toute équipe de développement doit s'y intéresser.

Cependant, il m'a fallu prendre en considération deux facteurs avant d'opter pour une solution pérenne et définitive :

- l'effort de recherche, de test et de validation lors du choix du VCS à utiliser, pour s'assurer qu'il réponde aux critères de choix (facilité d'utilisation, support du vendeur, prise en charge des fichiers de code) et qu'il puisse s'intégrer facilement dans le système d'information de Lilly (pare-feu, gestion des accès distants,…).
- le principe d'une solution VCS est de pouvoir être utilisable par tous les membres de l'équipe projet participant au développement. Ceci sans risque de mauvaise utilisation et sans possibilité de contourner les principes d'unicité de version et d'accès au fichier. Ce qui implique un effort pour s'assurer que chaque analyste comprenne comment fonctionne cet outil et que chacun l'utilise de la même manière.

Ces deux facteurs m'ont donc indiqué que l'utilisation d'un logiciel VCS requerrait un effort supplémentaire de la part de l'équipe projet et un risque que son utilisation ne soit pas maîtrisée par tous les membres de l'équipe.

IV.1.1.5 Mise en place d'un nouveau processus du suivi de fichiers

Au vu de cette analyse, et du fait du nombre restreint de développeurs analystes, mon choix a été de me baser sur la méthode utilisée par l'équipe Global MI&CS Lab, tout en l'améliorant, de telle manière à ce que les points de contrôle, entourés en bleu dans la figure '24-Processus de base de gestion des fichiers de code', puissent être assurés par une seule personne, ceci afin de limiter le risque d'erreurs lors des sorties et des réintégrations de fichiers.

La mise en place du processus a eu lieu lors du démarrage du développement, avec les ressources issues de l'équipe de support offshore. Il s'agissait donc de personnes au fait de la méthode employée par l'équipe Global MI&CS Lab.

Cependant, travaillant avec ces ressources éloignées depuis plus de deux ans, je connaissais très bien les possibilités d'erreurs d'interprétation, les étourderies et les changements de ressources fréquents. C'est pourquoi aucune amélioration n'a été apportée aux étapes dédiées au développeur-analyste. Seul le rôle du gestionnaire de fichier a vu le nombre de tâches augmenter.

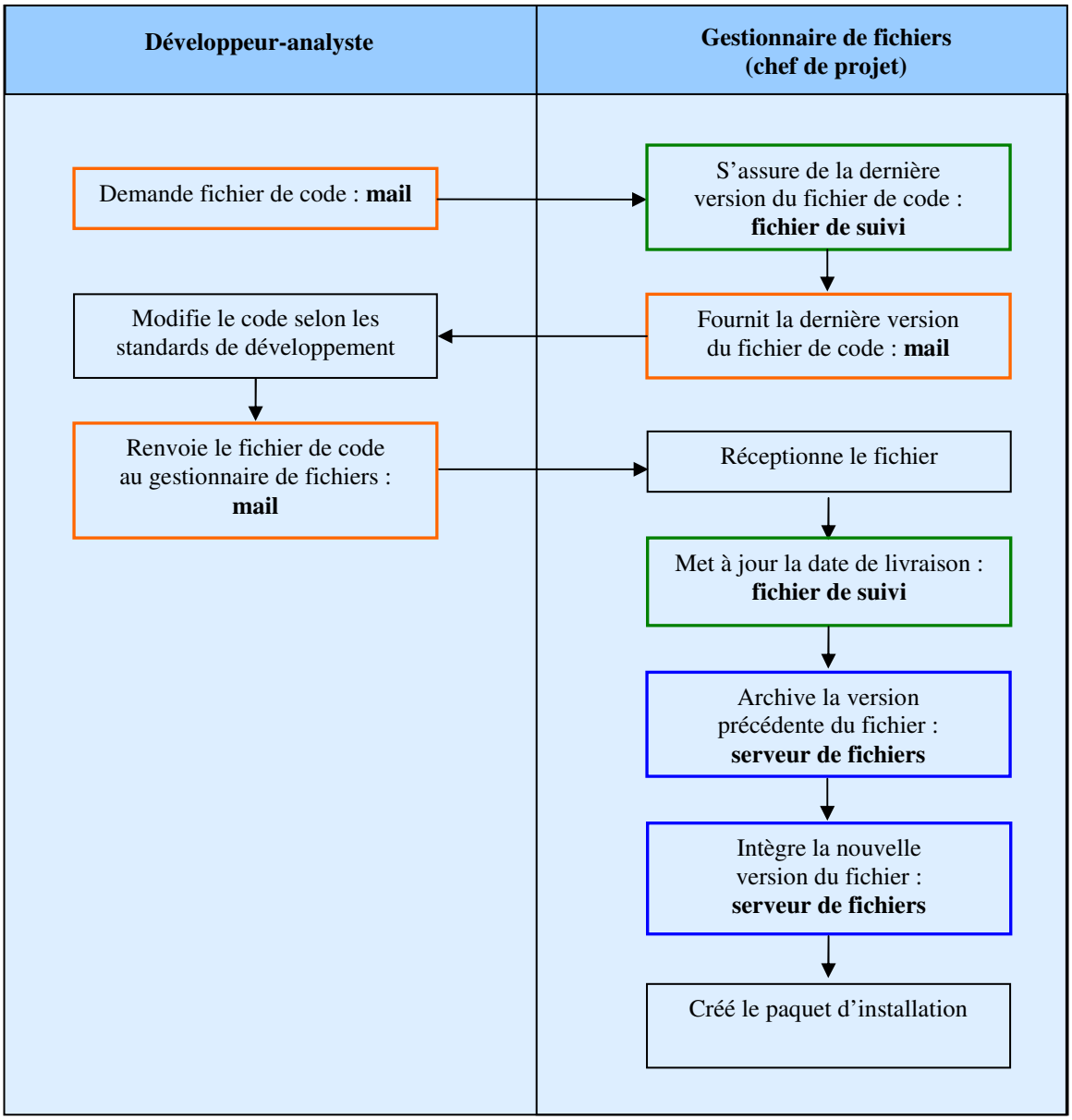


Figure 25 - Schéma du processus de gestion des fichiers de code – projet BOLDER R3.1

Comme on le voit dans la figure ci-dessus, j'ai choisi l'utilisation d'outils électroniques de base afin d'enregistrer précisément les actions effectuées, à des fins de contrôle.

Ainsi, j'ai plusieurs fois refusé d'intégrer des fichiers de code qu'un développeur avait modifié sans m'en faire la demande. Ces fichiers se trouvant sur le serveur applicatif, rien ne m'indiquait que d'autres modifications n'avaient pas été faites entre-temps. De plus, l'absence de demande et d'envoi par mail ne me permettaient plus de tracer le mouvement des fichiers.

En cas de besoin de développements concurrents sur le même fichier, j'ai privilégié le travail en série. Le deuxième développeur recevait ainsi le fichier à modifier une fois le testing unitaire concluant et les premiers travaux finis. Dans le cas où les modifications étaient minimes, je demandai alors au deuxième développeur de transmettre les lignes de code à insérer au premier.

Cette discipline et ces moyens de contrôle m'ont permis de garantir l'unicité des fichiers de code développés et d'éviter des pertes de temps.

Avec finalement très peu de moyens techniques, mis à disposition dans le système d'information de Lilly, j'ai pu mettre en place un système de suivi efficace : dans la mesure où j'ai effectué l'essentiel des actions de suivi, le nombre d'erreurs a été limité et celles-ci ont pu être corrigées très rapidement.

IV.1.2 Méthodes de travail

IV.1.2.1 Importance du respect du modèle de la base de données

Les premiers développements d'une centaine de fichiers de code d'interface a été effectué au courant de l'été 2011 par l' équipe offshore à New Dehli. L'éloignement géographique, ainsi que le niveau de qualité inégal dans les développements, a posé le

problème du respect du modèle de la base de données BOLDER, créé et validé en collaboration avec l'expert technique Lilly. Ainsi les développements de fichiers de code Ada!Rec ont pu contenir des erreurs quant à la définition des champs des bases de données source ou cible, ou des omissions de clés primaire, par exemple, bien que le testing unitaire aurait dû été effectué pour chaque développement. Ces erreurs auraient été découvertes lors des premiers tests du paquet d'interface, mais auraient alors rajouté des corrections, assorties d'autres tests unitaires et de revues.

Pour pallier à ce problème, je me suis donc attaché à revoir de manière systématique tout fichier de code qui m'a été fourni dans les premiers mois. Ceci afin de m'assurer que le développement suivait bien ledit modèle.

Afin de m'assurer que le code répondait bien aux besoins exprimés et aux spécifications techniques, j'ai effectué cette revue en suivant le modèle du cycle en 'V'. Je me suis appuyé sur les documents décrivant les requis de l'application BOLDER R3.1 et le modèle de la base de donnée, créée quelques mois plus tôt.

Chaque revue a donné lieu à des commentaires techniques précis (ligne par ligne), mais aussi concernant le respect des normes de développement en vigueur chez Lilly. On trouvera ainsi en annexe 7 un extrait de revue de code Ada!Rec qui s'assure que les normes sont correctement suivies.

IV.1.2.2 Systématisation des tests unitaires

Lesdites normes de développement imposent un test unitaire pour chaque fichier de code produit. Je me suis donc assuré que les développeurs testent informellement leur code à la fin de leurs travaux avant de pouvoir intégrer les fichiers de code au paquet d'installation.

Par ailleurs, à partir du mois de décembre 2011, l'entièreté du paquet d'installation a été testée à chaque nouvelle version dudit paquet, afin de s'assurer notamment du bon fonctionnement des scripts Unix qui lancent les scripts Ada!Rec et Oracle.

Il m'a fallu tenir compte du fait que plusieurs développeurs pouvaient travailler en parallèle, mais aussi du fait que BOLDER R3.1 va accueillir les données en provenance de six instances différentes. D'où un intérêt d'avoir à disposition plusieurs environnements de développement et test, avec chacun une source de données différente.

En m'appuyant sur les groupes de support des infrastructures et sur les ressources informatiques disponibles dans l'équipe Global MI&CS Lab, j'ai pu coordonner la mise en place de trois environnements de développement, comprenant la structure Unix et la base de données Oracle associée.

Enfin, la diversité des sources de données m'a amené à organiser un testing unitaire du paquet d'installation sur les copies des données de production de chaque instance EuroLIMS.

Ce testing s'est effectué au mois de juillet 2012, à la fin des travaux de développements : j'ai demandé l'autorisation de copier la dernière version des données de production aux représentants de chaque site. Puis, une fois la copie effectuée par l'équipe offshore, les jeux de données ont été installés tour à tour, afin de détecter les erreurs lors des transferts globaux et incrémentaux.

IV.1.2.3 Nécessité d'analyse en profondeur

Comme il a été indiqué, BOLDER R3.1 va contenir les données de six instances EuroLIMS différents, pour une période écoulée de quinze ans.

Chacune de ces six instances a été utilisée de manière différente par chaque site, selon les processus métiers en vigueur. Certains modules ont pu être utilisés, puis devenir inactifs, ou la structuration des données a pu changer entre la Release 1 et la Release 2 d'EuroLIMS.

J'ai donc à chaque fois demandé aux analystes d'étendre leur périmètre d'analyse au-delà de l'instance testée, copie d'un seul site. Et j'ai systématisé cette approche afin de s'assurer que les développements prennent en compte les cas d'exception dans l'utilisation des données selon l'instance et l'époque.

Nous trouverons ci-après trois exemples pour illustrer cette démarche :

- **L'utilisation des audits de données dans le temps**

EuroLIMS comporte un module d'audits de données, qui consiste à enregistrer les actions faites sur celles-ci et leurs valeurs modifiées. L'audit s'active par table pour les actions d'insertion, de modification et de suppression et aussi par champ pour les actions de modification.

Un développement spécifique a été conduit pour assurer le transfert des données d'audits et leur rattachement aux données auditées.

Dans la mesure où chaque site a pu utiliser les audits de manière différente à la fois dans le temps et en comparaison des autres sites, chaque problème rencontré lors du testing unitaire sur une copie d'instance a donné lieu à une analyse sur l'ensemble des autres copies de bases de données, afin de confirmer les différences d'utilisation.

C'est ainsi qu'il a été découvert que la méthode de rattachement de certains audits aux données auditées dans EuroLIMS avait changé entre la version EuroLIMS R1 et la version EuroLIMS R2.

Ce qui a permis de corriger le code concerné dans la phase de développement. Avec cette décision de testing systématique, nous avons pu éviter de rencontrer les mêmes problèmes lors des implémentations en production.

- **L'évaluation des volumes à transférer**

Lors des tests unitaires effectués au courant du printemps 2012 sur la copie d'instance EuroLIMS de Fegersheim, le transfert complet des données a échoué pour faute de place suffisante sur le serveur Unix. Ceci a posé la question de la place nécessaire lors des futurs déploiements en production, qui demanderaient forcément plus de place.

Afin de préparer ceux-ci, j'ai donc demandé aux analystes d'effectuer des tests unitaires du paquet d'interface, en utilisant une copie récente de chaque instance de production d'EuroLIMS.

Lors de ces tests unitaires du paquet d'installation, j'ai demandé à noter entre autres les informations suivantes pour chaque instance lors des tests unitaires :

- les temps de transfert,

- le volume total de données transférées dans les fichiers plats,
- la taille nécessaire pour les tables Oracle BOLDER, pour le stockage temporaire (TEMP_01) ou d'annulation (UNDO).

Le but de cette démarche étant de détecter quels paramètres devront être demandés lors de la création de la base de données ou du compte Unix lors des installations en production, mais également pour prévoir le temps d'installation.

J'ai aussi demandé la mise à jour du protocole servant à l'installation de la structure de BOLDER, afin de prévoir des paramètres de place suffisamment large lors des futures installations en production.

Les valeurs demandées dans ce protocole ne seront pas spécifiques à chaque instance de production, mais sont plutôt des valeurs convenant au plus d'instances possibles. Ceci étant assorti de recommandations d'adapter ces valeurs selon le volume de données à transférer depuis EuroLIMS.

Comme on le voit dans le tableau ci-dessous, le ratio entre les données de la base EuroLIMS et le volume des fichiers plats créés varie entre un pour trois et un pour cinq.

Tableau 8 : récapitulatif des volumes à transférer – instances au mois de Juillet 2012

	Alcobendas	Basingtoke	Fegersheim	Kinsale	Sesto	Speke
Volume de la base EuroLIMS en Giga Octets	5.6	1.6	15.3 (*)	6.5	5.9	2.2
Volume des fichiers plats à transférer en MO	20.1	8.8	40	17.8	18.8	10.2

(*) copie de Mars 2011.

- **Implémentation des filtres dans l'univers Business Objects**

L'univers Business Objects de la version BOLDER R3.0 contient un certain nombre de filtres, objets créés par les développeurs de l'univers et qui permettront aux créateurs des rapports de réduire facilement le champ de leurs recherches en utilisant ces filtres.

Ainsi, lors de la recherche d'échantillons de laboratoire pour un produit précis, seuls ceux en statut 'Approuvé' peuvent être demandés par l'utilisateur. En mettant à disposition une série de filtres basés sur les différents statuts des échantillons dans EuroLIMS, on permet à l'utilisateur de gagner du temps lors de la création de son rapport.

Pour des raisons de cohérence et continuité vis-à-vis de la version BOLDER R3.0 et pour présenter une application de reporting qui propose des fonctionnalités utiles, il a été décidé en juillet 2012 de rajouter des filtres pour un certain nombre de données de BOLDER R3.1.

Cependant, en raison du nombre important de données rajoutées et du temps de développement nécessaire, j'ai demandé de recenser tous les filtres potentiels dans l'univers. Cette analyse s'est faite en parallèle des autres travaux de développement Business Objects, afin de ne pas pénaliser l'avancée de ceux-ci.

Le nombre de filtres potentiels s'est élevé à huit mille. Leur temps de développement s'est donc révélé trop important en regard de l'avancée du projet, et de la contrainte de livraison.

Le travail d'analyse détaillée a alors servi à déterminer précisément quels filtres seraient réellement utiles dans l'application. Une fois ce choix fait, le nombre de filtres potentiels est passé à sept cents.

En plus d'une exploitation immédiate d'une partie des résultats de ces analyses techniques, ceux-ci ont à chaque fois été archivés dans le serveur de fichiers dédié au projet BOLDER R3.1, afin de pouvoir les exploiter ultérieurement. J'ai systématisé cette démarche, en gardant à l'esprit les développements potentiels dans d'autres versions de BOLDER.

IV.1.3 Cas de recherche et développement

La plupart des données d'EuroLIMS à transférer dans BOLDER R3.1 proviennent d'une base de données relationnelle Oracle, et sont enregistrées dans des champs de type texte, date ou numérique. L'outil de transfert développé peut donc migrer les données EuroLIMS comme attendu. Il existe cependant deux autres types de données, qui sont stockées de deux manières différentes.

Par exemple, EuroLIMS propose un module de calcul de résultats, qui permet aux utilisateurs d'utiliser des formules de calcul élaborées sur demande.

Le code utilisé est enregistré dans des fichiers sur le serveur applicatif Unix.L'accès à ce code se fait en ouvrant la fiche descriptive de la formule de calcul ('Calculation').

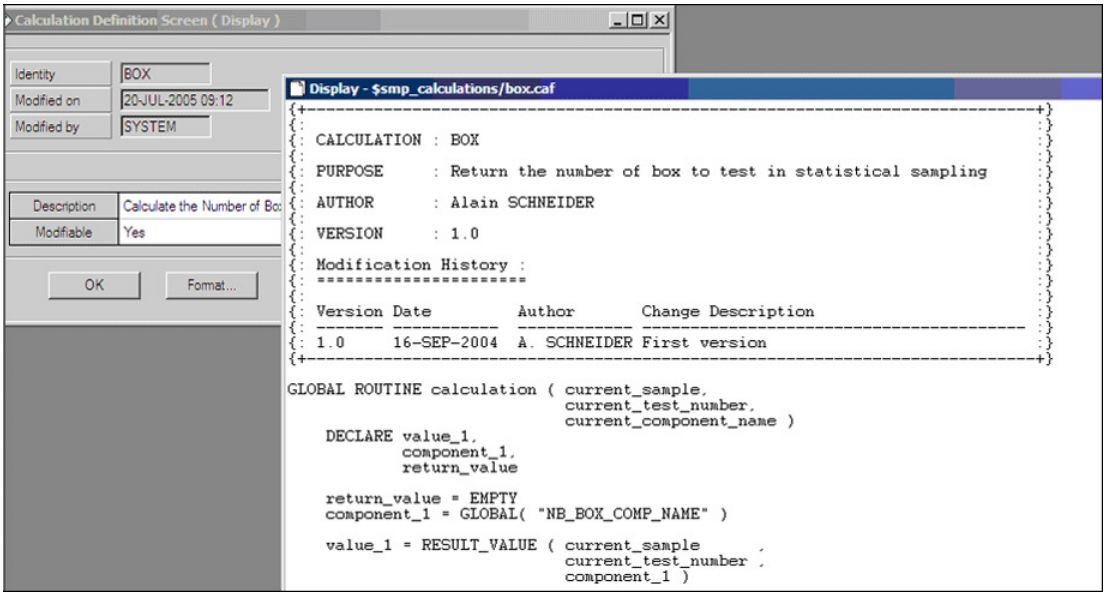


Figure 26 - Extrait de l'application EuroLIMS : fichier de code utilisé pour calculer des résultats complexes.
En cliquant sur le bouton 'Format..', l'utilisateur affiche le code utilisé pour calculer des résultats complexes, stocké sous forme de fichier texte.

Dans d'autres cas, les données compilés sont stockées dans des champs de type BLOB[25]. Il s'agit par exemple de données graphiques, qui constituent des impressions virtuelles de certificats d'analyse.

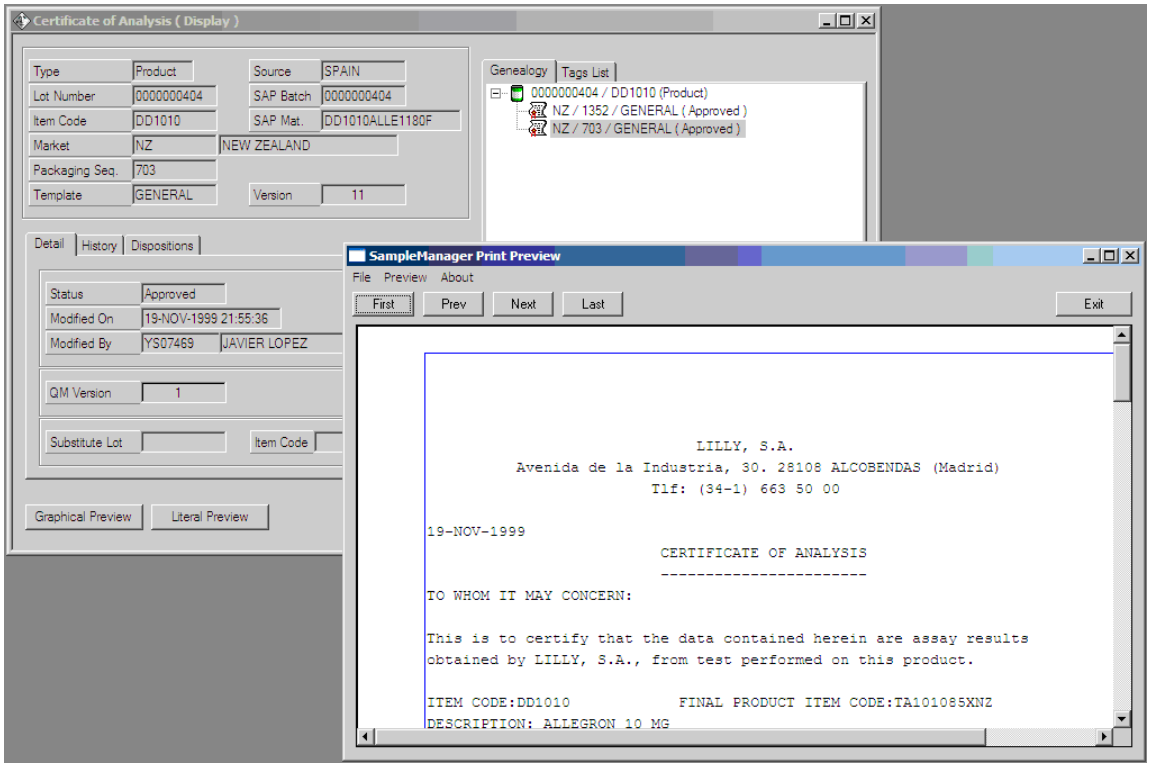


Figure 27 - Extrait de l'application EuroLIMS : données stockées en BLOB, présentées en impression virtuelle.
En cliquant sur le bouton 'Graphical Preview..', l'utilisateur affiche le un jeu de données stockées sous forme de BLOB et présentées en impression virtuelle.

[25] Binary Large Object : type de champ de base de données permettant de stocker des données binaires

Le processus standard de transfert ne permet pas de migrer les données stockées dans des BLOB ou des fichiers plats stockés sur le serveur Unix.

Un des objectifs du projet a donc été de développer un processus spécifique pouvant extraire ces données particulières de la structure d'EuroLIMS, les transférer et les intégrer dans l'application BOLDER, ceci de manière récurrente.

Le tableau ci-dessous un récapitulatif de ces données particulières ainsi qu'une estimation de leur nombre par instance production. Les volumes indiqués montrent clairement qu'une opération manuelle n'est pas envisageable.

Tableau 9 : récapitulatif des fichiers et données de BLOB à transférer – instances au mois de Juillet 2012

	Alcobendas	Basingtoke	Fegersheim	Kinsale	Sesto	Speke
Nombre de fichiers plats	377	197	2 291	2 090	2 340	67
Nombre de données BLOB	80 017	47 900	199 834	35 000	120 500	74 700

La complexité du développement, la cohabitation de différents environnements techniques et l'absence de précédents connus ont impliqué un travail de recherche et de développement, afin de trouver une solution technique pour chaque partie de ce nouveau processus.

Ma stratégie a été de découper en tâches distinctes le processus d'extraction, transfert et intégration, afin de pouvoir travailler en parallèle sur plusieurs d'entre elles ou d'intégrer les travaux effectués de manière séparée.

Ainsi, nous pouvons retrouver les fonctionnalités suivantes :

1 - extraction des données d'EuroLIMS (routine en code VGL)

2- impression virtuelle (PDF) du contenu des fichiers

3- transfert des fichiers du serveur Unix EuroLIMS au serveur Linux BOLDER

4- stockage des fichiers PDF dans la base de données BOLDER, sous forme binaire (BLOB)

5- stockage des mêmes fichiers PDF sur le serveur Linux

6- accès pour les utilisateurs à ces fichiers via le système de fichiers Windows, l'application Samba permettant de gérer l'accès aux fichiers pour tout utilisateur ayant les droits idoines attribués par le système de droits de Windows.

Chaque segment a pu ainsi être développé et testé séparément. Ce qui a permis d'avancer sur ce développement de manière asynchrone, en profitant parfois de la disponibilité temporaire de ressources non allouées au projet.

En tant que chef de projet, j'ai demandé l'implémentation des différents outils nécessaires aux groupes de support concernés, utilisant ainsi ma connaissance de l'organisation informatique de Lilly. J'ai ainsi assuré l'intégration des différents développements afin de constituer l'ensemble du module.

A la fin de l'année 2011, le développement de ce dernier s'est arrêté, les travaux de recherche et développement ayant démontré certaines limitations techniques. Il a été décidé par le chef de service d'exclure de la version BOLDER R3.1 ce module transfert de données, et d'y revenir dans une version ultérieure, au moment du retrait d'EuroLIMS. En effet, ces données seront toujours disponibles dans EuroLIMS et il ne sera nécessaire de les transférer qu'à la fin de l'utilisation du SIGL.

Je me suis donc assuré que l'ensemble des fichiers de code, des documents techniques et logiciels de cette preuve de concept soient archivés dans une arborescence correspondant à la segmentation précitée.

Ceci pour mettre à disposition de la prochaine équipe de développement une base documentaire et technique permettant de reprendre rapidement les travaux de développement.

IV.2 Chapitre deux : développements techniques

IV.2.1 L'écriture des spécifications techniques

Comme déterminé lors de l'écriture du plan de projet, l'écriture des documents de spécifications techniques a été prévue comme une tâche parallèle aux travaux de développements. Ce qui a présenté un avantage certain en termes de souplesse de développement.

Cette démarche représente cependant un risque dans la mesure où le code est créé directement, sans faire revoir de suite le modèle technique par un pair. Le risque a été minimisé par la validation du modèle de données par l'architecte technique et par l'organisation de plusieurs revues intermédiaires des spécifications techniques.

IV.2.1.1 Confirmation directe des modèles

Chaque table BOLDER est alimentée avec les données EuroLIMS grâce à un jeu de fichiers Ada!Rec. Les développements ont pu donc être segmentés de manière fine, ce qui a facilité les travaux de recherche et développement. Le schéma de l'interface correspondante a été figé dans le document de spécification fonctionnelle, une fois le code développé selon les attendus et les tests unitaires concluants.

Cette méthode a permis de pouvoir travailler concrètement sur les problématiques techniques, et d'éviter de revenir à chaque modification du code sur les spécifications techniques.

Dans le cas où le code a dû être réécrit suite à correction, je me suis assuré que les analystes mettent à jour les documents techniques correspondants, en les mettant à leur disposition et en contrôlant, notamment au début, la mise à jour des documents.

IV.2.1.2 Une revue plus simple

Comme tous les documents de validation, les modèles techniques doivent être revus par une deuxième personne, ceci afin de s'assurer que le modèle proposé réponde bien au cahier des charges.

Les premiers développements et donc les premiers documents ont été écrits par les analystes, travaillant à distance.

En conséquence, afin de m'assurer que le contenu des documents soit correct dès le départ, j'ai tout d'abord effectué une revue régulière à chaque livraison du document de design.

A l'arrivée des deux ressources dans l'équipe de Fegersheim, le nombre de tâches à encadrer a naturellement augmenté. Par ailleurs, les documents ayant été revus et corrigés plusieurs fois, les nouvelles versions nécessitaient moins de relecture, les analystes ayant intégrés les méthodes de rédaction. J'ai ainsi ralenti mon rythme de revue.

La revue des documents de spécifications technique et fonctionnelle devrait reprendre une fois la phase de testing informelle exécutée.

Elle se fera en suivant le cycle de projet en V : en partant du cahier des charges, je reverrai et validerai les documents de spécifications technique et fonctionnelle, modifié ou créé pour cette version de BOLDER.

Dans le cas de documents non liés à des demandes du cahier des charges (ex : liens des tables dans l'univers Business Objects), il me faudra prendre le temps de comprendre le développement et de m'assurer que le document correspondant l'explique correctement.

Il est bien évident que je ne ferai pas cette revue seul, surtout au vu du nombre d'artefacts créés, et que l'ensemble de l'équipe projet devrait participer à cet effort, à la mesure de ses compétences techniques.

IV.2.2 Le développement de l'interface et de la base de données

Comme indiqué, je suis peu intervenu dans les développements du paquet d'interface et de la base de données.

J'ai laissé autonomie aux analystes pour développer le code nécessaire et écrire les spécifications techniques correspondantes. Mes quelques interventions ont surtout servi à prendre certaines décisions quant à l'utilité de tel ou tel développement, ou à aider à comprendre un problème technique.

Cette démarche a été rendue nécessaire du fait des autres activités de support et de formations qui se sont déroulé en parallèle des travaux du projet.

J'ai par contre été présent sur trois activités en rapport avec le développement :

- la gestion et le suivi des fichiers de code, selon le processus indiqué dans les pages précédentes. J'ai en effet décidé de ne pas déléguer cette activité (sauf congés) afin de pouvoir garder le contrôle de l'unicité des artefacts développés, pour éviter toute complication ultérieure, du fait de l'existence de deux fichiers du même nom par exemple.

- la relecture des documents de validation produits : spécifications techniques, protocoles d'installation, plan de test, guide de sécurité,… Mon expérience au sein du Global MI&CS Lab et la vue d'ensemble que je dois avoir en tant que chef de projet ont impliqué que je m'implique dans cette tâche.

- la mise à disposition d'environnements de développements et de tests : ma connaissance des équipes de support des infrastructures Lilly a contribué à accélérer la mise en place des éléments nécessaires, mais aussi à obtenir du support technique lors de problèmes techniques (droits manquants, volumes insuffisants,…).

IV.2.3 Les artefacts Business Objects

IV.2.3.1 un travail plus conséquent que prévu

Le développement de l'univers Business Objects a commencé en avril 2012. Très rapidement, l'estimation de temps de développement initiale a été revisitée, suite aux constatations faites par le consultant chargé de son développement.

L'ajout des nouveaux champs de la base de données a impliqué en effet au final de devoir rajouter un nombre conséquent d'objets dans l'univers.

Tableau 10 : récapitulatif des ajouts d'artefacts dans l'univers Business Objects

	BOLDER_3.0	BOLDER_3.1	Delta	% ajouté
Classes	232	618	386	62,46%
Objets	2561	7370	4809	65,25%
Tables	60	131	71	54,20%
Alias	166	497	331	66,60%
Jointures	220	595	375	63,03%
Conditions	203	226	23	10,18%
Sous univers	6	7	1	16.66%

Cette analyse permettant de visualiser la différence en termes d'objets s'est faite au fur et à mesure des développements et des points avec l'expert technique Lilly. Ceci en raison de l'analyse progressive des développements techniques nécessaires, mais aussi de précisions apportées aux besoins non exprimés dans le cahier des charges.

L'univers a ainsi fortement augmenté comme l'indique le tableau ci-dessus, mais il a aussi gagné en complexité, comme le montrent les chiffres pour les différents objets de l'univers : alias, jointures, conditions,...

J'ai pris en compte ces multiples changements d'hypothèse de développements, en réactualisant les charges prévisionnelles, en intégrant celle-ci à chaque fois dans le plan de projet et en informant le Global Custodian de l'avancée ralentie des travaux.

Ceux-ci se sont déroulés du mois d'avril au mois de juillet 2012, période à laquelle la priorité a été donnée à l'intégration du nouveau prestataire offshore. Les travaux du développement de l'univers se sont donc déroulés de manière autonome, l'analyste ayant à sa disposition un environnement de développement et de tests lui permettant d'exécuter son testing unitaire.

IV.2.3.2 une expertise technique et métier nécessaire

Comme indiqué en amont, les personnes Lilly disposant d'une expertise technique et fonctionnelle ont été peu disponibles tout au long du projet. Le développement de l'interface et de la base de données a pu cependant se faire sans qu'il soit nécessaire de faire appel à leurs connaissances, sauf cas particulier.

Le développement de l'univers a par contre clairement requis leur intervention. D'une part pour des raisons techniques : les liens possibles entre les tables BOLDER au sein de l'univers sont directement liés à la manière dont les données EuroLIMS sont interconnectées. De plus, chaque site a pu utiliser certains types de données de manière particulière, ou bien être le seul à le faire.

L'expertise fonctionnelle a aussi été requise pour déterminer comment l'application BOLDER R3.1 devrait présenter les informations à l'utilisateur : comment les objets seraient nommés, comment seraient-ils classés dans les sous-univers,…

Pour répondre à ce besoin d'expertise, une série de réunions de travail d'une demi-journée ont été organisées à mon initiative à intervalles réguliers entre l'équipe projet et les deux experts, afin de faire le point sur les développements effectués, d'ajuster ceux-ci et d'enlever ou de rajouter des éléments dans l'univers Business Objects.

Ce qui a permis de travailler de manière itérative et de progresser de manière continue dans le développement de l'univers.

Une fois cette méthode de travail mise en place, j'ai laissé le développeur de l'univers Business Objects travailler directement avec les deux experts, le laissant organiser d'autres rencontres et en m'assurant qu'il recevait bien l'ensemble des informations dont il avait besoin. Je suis revenu vers lui pour estimer l'avancée des travaux, les développements restants, et de voir avec lui comment une deuxième ressource pouvait lui prêter main-forte sur l'analyse des travaux à effectuer.

V Cinquième partie : préparation des tests et étapes à venir

[Juillet – Août 2012]

Les travaux de développement se sont achevés à la fin du mois de juillet de cette année. Quelques ajustements ont encore été effectués sur l'univers, et des rapports Business Objects servant à tester les principes fonctionnels de l'application seront développés au courant du mois d'août.

V.1 Chapitre un : accélération du projet

Au début de mois de Juillet, la phase de transformation de l'équipe Global MICS Lab[26] est passée en phase de stabilisation. Ce qui a signifié que les ressources allouées à cette phase ont pu revenir vers leurs activités d'origine, et que les projets, mis en attente ou au second plan, ont été remis progressivement au premier plan.

Dans le même temps, la contrainte de temps est redevenue réelle. Le Global Custodian a clairement signifié lors d'une réunion de service que la date de livraison de l'application devrait être avancée, pour des raisons externes au projet.

La problématique a alors été pour moi de déterminer comment raccourcir le chemin critique du projet, constitué notamment des actions de test informels et formels.

Ce qui a posé de suite la question du temps qui devrait être consacré au testing informel, phase pendant laquelle l'ensemble des artefacts doit être testé.

Dans cette phase, le déploiement de l'environnement de test se fera selon les protocoles écrits pour la version R3.1, le paquet d'installation et l'univers seront testés afin de vérifier l'absence de bogues systèmes ou d'erreurs documentaires. Il sera également vérifié que les requis du cahier des charges sont satisfaites mais aussi d'autres demandes, exprimées par les experts technique et métier, dans les spécifications fonctionnelles.

La problématique fut pour moi de trouver comment accélérer la phase de testing informel et la rendre plus efficiente.

[26] Décrite dans la deuxième partie, chapitre un section 4 'Les contraintes externes »

J'ai suivi deux pistes en parallèle : l'augmentation de ressources et l'exécution d'une première série de tests avant la fin des travaux sur l'univers.

V.1.1 Renfort de ressources

Le modèle de service du nouveau prestataire a permis à Lilly de renforcer le nombre de ressources entièrement dédiées au projet.

Dans ce cadre, j'ai pu ainsi demander l'appoint de trois ressources allouées au testing.

Ce nombre a été défini sur la base de deux critères :

- un renforcement de l'équipe de testeurs accélèrera bien évidemment l'exécution du testing
- cependant, un trop grand nombre de nouvelles ressources pourrait être contre-productif : elles devront être encadrées, car débutant à la fois sur l'application BOLDER et vraisemblablement sur la validation pharmaceutique. La solution serait alors de leur attribuer à chacune d'entre elles un 'tuteur', qui les accompagnerait dans leurs tâches de testing et les reverrait en détail. Ce rôle sera assuré par les deux analystes présents à Fegersheim, et moi-même.

J'ai défini les profils techniques demandés, en définissant la date de disponibilité souhaitée et en précisant qu'au moins une des ressources devait avoir des compétences en développement.

Dans les deux profils, les compétences techniques (Ada!Rec, SQL, Unix, Business Objects), applicatives (EuroLIMS et BOLDER) et en validation restent les mêmes. La différence se fait sur l'expérience et le niveau technique : le profil de développeur implique de plus grandes connaissances techniques, afin d'être à même de reprendre les développements techniques en cas de corrections.

En effet, le contrat des deux ressources projet présentes à Fegersheim était prévu, au moment lors de cette demande de renfort, de s'arrêter à la fin du testing informel. En cas de corrections nécessaires sur le paquet d'installation ou sur les artefacts Business Objects, une ressource capable de reprendre les développements devrait être disponible.

Notons par ailleurs l'avantage d'avoir un développeur ayant participé au testing : la montée en connaissance des applications EuroLIMS et BOLDER se fera tout au long du testing et la compréhension technique des bogues éventuels sera facilitée.

V.1.2 Avancer l'exécution des tests

Dans le plan de projet initial, le début du testing était lié par une contrainte 'Fin-Départ' à la fin du développement.

La deuxième piste pour accélérer la phase de test a donc été de lancer le testing informel avant la fin des travaux sur l'univers Business Objects.

La très grande majorité des tests (plus de 75%) concernent la vérification du bon transfert des données demandées dans le cahier des charges, comme on peut le voir dans le tableau ci-dessous.

Tableau 11 : récapitulatif des totaux de tests par types

Type de test	Description	Nombre	Pourcentage sur total
BOU	Rapports	4	1,93 %
DAT	Transferts de données spécifiques	8	3,87 %
ERS	Enregistrements électroniques et sécurité des données	4	1,93 %
REP	Informations génériques des rapports	1	0,48 %
SEC	Sécurité des éléments de l'application BOLDER	5	2,42 %
SYS	Système	2	1,24 %
STR	Tests des rapports Business Objects	22	10,67 %
TRN	Transfert de données	160	77,66 %
	Total de tests	206	100 %

Un exemple type de ces tests de transferts de données consiste :

- à créer un jeu de données dans EuroLIMS,

- puis à lancer l'outil de transfert incrémental

- et enfin à vérifier dans BOLDER à l'aide de requêtes SQL si les données ont été correctement transférée, puis en utilisant un rapport Business Objects pour tester si elles s'affichent correctement.

Ce test dit ‘d’insertion’ est suivi d’un test où il est vérifié si les mêmes données modifiées sont bien transférées. Il existe certains cas où l’on teste la suppression des données.

V.1.3 Le nouvel ordonnancement des tests

Mon idée a donc été d’estimer quelles actions de test pourraient être effectuées sans devoir utiliser les rapports Business Objects cités.

J’ai donc évalué le travail que représenterait une séparation des étapes de tests, pour ne travailler dans un premier temps sur la validation du transfert de données.

Les jeux de données seraient donc mis en place et les premières étapes de test réalisées jusqu’avant celle impliquant l’utilisation d’un rapport Business Objects. Le test serait alors mis en pause, et un autre serait exécuté, ceci jusqu’à épuisement des tests de transfert de données.

Les étapes de chaque test impliquant le rapport Business Objects dédié seront alors exécutées une fois les artefacts Business Objects disponibles (univers et rapport).

Le lecteur comprendra rapidement les avantages de cette approche :

- le testing informel pourrait donc commencer sans attendre la fin des travaux de l’univers, ce qui éviterait des creux d’activités pour l’équipe projet de Fegersheim, les ressources demandées au prestataire offshore intégrant plus tard la phase de tests.

- les travaux sur l’univers Business Objects, puis sur les rapports de tests pourront se dérouler en parallèle du début du testing informel, sans qu’il n’y ait une trop forte pression sur le consultant chargé de leur développement.

- en effectuant les tests de transferts et d’intégration de données en trois temps, mais en les conservant au sein du même test, j’évite ainsi une modification des tests concernés et une nouvelle rédaction de cas de tests.

Afin de valider cette approche, j’ai effectué un état des lieux de l’ensemble des tests existants et ai noté pour chacun d’entre eux quelles étapes pouvaient être faites sans utiliser Business Objects.

Je me suis également assuré que les tests retenus pour cette première phase de test n'étaient pas liés entre eux ou l'étaient de telle manière qu'une exécution en parallèle était possible.

Il est bien évident que cette approche nécessite une préparation détaillée du déroulé des tests et un suivi rigoureux de celui-ci par les différents analystes. Une série d'instructions devra être communiquée à chacun d'entre eux avant le début de leurs activités de testing, et je devrai m'assurer que les tests soient effectués étape par étape.

Dans le cas où les ressources supplémentaires ne seront pas disponibles, une autre stratégie de test devra être mise en place, qui prendra en compte le nombre restreint de testeurs.

Grâce à cette analyse, j'ai pu établir un plan d'exécution des premiers tests, qui comprend à la fois la mise en place des jeux de données et les premières étapes.

Afin de pouvoir obtenir une estimation d'exécution moyenne, j'ai demandé à deux ressources de niveau d'expertise différente de dérouler trois tests ciblés, afin d'évaluer les temps d'exécution pour une ressource chevronnée et pour une ressource débutante.

J'ai pu estimer pour chacun des tests (environ deux cents) le temps d'exécution pour un testeur expérimenté basé chez Lilly à Fegersheim (proche des instances BOLDER) et pour un testeur situé à l'étranger (Inde ou EUA) se connectant au réseau Lilly via un réseau tiers et une machine virtuelle.

	A	B	C	D	E	F	G
1	Test Numbe	Purpose	Transfer ?	Step for transfer	BO ?	Time estimated (hours senior Feg)	Time estimated (junior India)
49	IT.TRN.0.0	Prep of Full transfer	NO		NO	4	6
50	IT.TRN.0.1	Full transfer	FULL	3	NO	8	10
51	IT.TRN.0.2	Full transfer	FULL	5	NO	8	10
52	IT.TRN.0.3	Full transfer	FULL	5	NO	8	10
53	IT.TRN.0.4	Full transfer	FULL	5	NO	10	12
54	IT.TRN.0.5	Full transfer	FULL	5	NO	8	10
55	IT.TRN.0.6	Full transfer	FULL	5	NO	8	10
56	IT.TRN.0.7	Full transfer	FULL	5	NO	8	10
57	IT.TRN.0.8	Full transfer	FULL	5	NO	8	10
58	IT.TRN.1.1	Ana+Comp	INC	5	YES	1	1,5
59	IT.TRN.1.2	Ana+Comp	INC	5	YES	1	1,5
60	IT.TRN.1.3	Ana+Comp	INC	4	YES	1	1,5
61	IT.TRN.1.4	PL	INC	4	YES	1	1,5
62	IT.TRN.2.1	READ only	NO		YES	0,5	1

Figure 28 - Extrait du tableau d'estimation des temps d'exécution par test

Les temps ont été estimés en prenant les cas les plus probables, mais qui présentaient également les valeurs les plus extrêmes : un testeur chevronné en France versus un testeur junior en Inde sur un réseau informatique distant. Ceci afin d'obtenir un différentiel de temps

d'exécution le plus large possible, afin de me ménager une marge de manœuvre lors du testing.

Cette estimation globale m'a ensuite permis d'estimer le volume total de jours d'exécution pour un testeur junior à soixante-dix jours. J'ai volontairement pris cette mesure haute, sachant que les tests les plus complexes et les plus longs seraient exécutés à Fegersheim par un testeur expérimenté. Le gain de temps potentiel a été mis en place pour garder une marge de sécurité lors de l'exécution générale.

Comme on peut le voir, ce tableau comporte également des informations concernant le contenu du test : y-a-il un transfert de données (incrémental ou complet), si oui, à quelle(s) étape(s) du test et le test demande-t-il de consulter un rapport Business Objects ?

Ce qui devrait permettre par exemple d'effectuer en série les premières étapes de tous tests demandant un transfert complet, puis de lancer celui-ci une seule fois, permettant ainsi un gain de temps.

Le testing devrait commencer avec les tests les plus longs effectués par un testeur expérimenté mi-août, puis avec les ressources offshore demandées arrivant en deux temps : une ressource et demie à mi-août et une ressource supplémentaire à fin août.

J'ai donc pu établir une date fin estimée de cette première étape de test à mi-septembre. Ceci impliquant également la présence de relecteurs de tests et d'analyses d'incidents, tâches qui seront effectuées en parallèle et qui n'ont pas été comptées dans l'estimation : mon idée est d'attribuer ce rôle aux analystes expérimentés basés à Fegersheim et moi-même.

V.1.4 Finalisation de la documentation de testing

Le testing informel est une étape cruciale dans le développement d'un projet informatique : il permet de tester et valider les développements, les tests et les documents créés, tout en s'affranchissant des revues et approbations formelles. C'est lors de cette étape

que l'équipe projet découvre le nombre de corrections à effectuer et prépare ainsi le futur testing formel.

Il est donc important de préparer cette étape avec soin, en s'assurant que les documents nécessaires soient revus et prêts à être utilisés.

En plus de la revue des cas de tests qui ont été migrés ou créés dans Quality Center, deux documents ont été achevés avant de commencer le testing :

- **Plan de test général ('Overall Test Plan') :**

Il s'agit du document qui décrit comment se déroule le testing de l'application BOLDER. Il n'est pas dépendant d'une version de l'application.

Je l'ai fait modifier par les analystes travaillant sur la migration des tests, pour intégrer la migration de l'ensemble des tests dans l'application Quality Center et il s'est vu doté d'un addendum qui décrit comment y créer et faire approuver des jeux de tests, comment les exécuter et revoir leurs exécution, et comment gérer les tests échoués.

J'ai achevé sa revue et ai soumis ce document pour approbation au courant du mois de juillet.

- **Plan de test de la version ('Release Test Plan')**

Ce document est propre à la version de l'application en cours de développement. Il décrit le contexte du projet et présente la stratégie de testing ainsi que l'ensemble des tests qui vont être exécutés. Seule son approbation informelle est requise avant le début du testing, ce document pouvant éventuellement être corrigé avant l'exécution du testing formel.

J'ai pu livrer une première version de ce document une fois l'ensemble des tests revus et choisis pour le testing informel, à la fin du mois de juillet. Une version définitive devra être approuvée avant le début du testing formel.

V.1.5 Préparation de l'environnement

Le testing d'une application implique une installation de l'environnement de test en déroulant des protocoles de déploiement revus et approuvés. Dans le cas du testing informel, ils peuvent être approuvés informellement si le but dudit testing est également de valider la bonne écriture de ces protocoles.

J'ai donc demandé à ce que soit installée sur une instance EuroLIMS de test une réplication de la base de production du site d'Alcobendas. Il s'agit en effet d'une des deux instances de production ayant utilisé, au cours de la vie d'EuroLIMS, tous les types de données étant transférés dans BOLDER R3.1. Et elle présente l'avantage d'être plus petite que l'autre instance (Fegersheim), donc de permettre de gagner du temps lors des copies ou transferts de données.

A cette instance EuroLIMS sera ensuite associée l'instance de test BOLDER, déployée en utilisant les nouveaux protocoles de déploiement. Ceux-ci seront corrigés le cas échéant et validés formellement avant le début du testing formel.

V.2 Chapitre trois : la suite du projet

V.2.1 Prochaines étapes

Les nombreuses contraintes documentaires, le volume des données à transférer, les nombreux artefacts créés indiquent les tâches qui me restent à accomplir pour les tâches restantes de testing et de validation.

Au moment où j'écris ces lignes, le testing informel d'intégration est donc en train de se mettre en place. Son exécution devrait démarrer mi-août 2012, puis être suivi d'une phase de correction des fichiers de code, des tests et des protocoles d'installation.

Cette phase sera suivie par la formalisation de l'ensemble des artefacts : revue et signature formelles des revues de codes, des tests, des documents de design et validation.

Ensuite commencera le testing formel d'intégration, avec une éventuelle phase de correction suivie d'une deuxième passe de tests.

Enfin viendra la clôture des travaux de validation, de la livraison de l'application et des formations.

Il est bien évident que nous ne connaissons pas encore le déroulé détaillé de ces futures étapes, mais nous pouvons déjà envisager certaines actions à venir :

- dans la mesure où la contrainte de temps a été accrue, l'objectif pour moi va être d'organiser en parallèle du chemin critique, constitué de la fin du développement et du testing, les actions restantes de corrections et de revues.
- les déploiements de la nouvelle application sur chaque site devront prendre en compte les développements spécifiques faits à leur demande à différentes époques. Il sera évalué si la version BOLDER R3.1 répond aux besoins exprimés alors, dans le cas d'ajouts d'objets Oracle par exemple (les clichés notamment).
- l'univers ayant beaucoup évolué, les documents de formations devront refléter ces changements et les expliquer aux utilisateurs actuels mais également prévoir l'utilisation future de BOLDER par des personnes non formées à EuroLIMS. Ces documents de formation devraient donc être créés en collaboration avec l'analyste responsable du développement de l'univers.
- le grand nombre de tâches à accomplir, la nécessité de devoir suivre leur avancée et de reporter au plus près la fin prévue des travaux, va impliquer que je délègue une partie de la coordination à un analyste offshore, afin de faciliter l'avancée des travaux sur place.
- enfin, je devrais continuer à documenter l'avancée du projet, en suivant en détail l'évolution de l'exécution des actions de revue et de validation, afin de réajuster régulièrement les dates de fin dans le plan de projet.

  Ces informations devront être régulièrement remontées vers le client Lilly, mais également vers la société prestataire qui emploie les ressources offshore.

V.2.2 Préparer la future version de BOLDER R3.2

Comme nous l'avons vu dans la section 'IV.1.3 - Cas de recherche et développement', un certain nombre de demandes exprimées dans le cahier des charges ont été retirées du projet, ceci en accord avec le Global Custodian. Ces demandes sont trop complexes à développer et ajoutent un risque supplémentaire pour ce projet.

Les travaux de développement de l'univers ont également montré un nombre conséquent d'améliorations potentielles, notamment l'implémentation de nouveaux filtres ou d'instruction de décodage.

La mise en production de la version BOLDER R3.1 révèlera peut-être des bogues ou des limitations qui devront être corrigés. Enfin les sites utilisateurs demanderont sans doute des améliorations ou des développements spécifiques une fois la nouvelle application maîtrisée.

C'est pourquoi une version 3.2 de BOLDER est déjà prévue. Ses travaux préliminaires démarreront vraisemblablement à la fin des étapes de validation, une fois la version BOLDER R3.1 finalisée et prête à être livrée.

Il me faudra alors rassembler les informations techniques collectées lors du projet, qui seront nécessaires au développement de la future application BOLDER R3.2, afin de pouvoir les transmettre à la future équipe projet.

Conclusion

J'ai présenté à travers ces pages les démarches suivies, les choix que j'ai pu faire en tant que chef de projet mais également les situations que j'ai dû accepter et intégrer dans le déroulé du projet.

L'analyse de problèmes rencontrés lors de l'avancée du projet, les solutions que j'y ai apportées participent d'un apprentissage.

Ainsi, afin de conclure ce mémoire, dont la rédaction s'est achevée sur la situation en août 2102, je vais consacrer quelques lignes aux leçons que j'ai pu retenir de ces activités de chef de projet et faire un bilan non-technique sur ce projet encore en cours, mais aussi tirer des enseignements pour ma propre expérience professionnelle.

- **Bilan sur le projet**
  - Des requis exprimés tardivement

Le projet BOLDER R3.1 a été entamé avec un cahier des charges basé très majoritairement sur le transfert de données d'EuroLIMS vers BOLDER. Il ne présente que très peu de requis fonctionnels.

Ce manque d'informations s'est ressenti bien après, lors du développement de l'univers Business Objects. C'est alors que des requis manquants ont été exprimés au fur et à mesure des travaux, lorsque les ressources expertes étaient disponibles. Aujourd'hui encore, au vu du volume de l'univers, il est attendu que d'autres requis fonctionnels doivent pris en compte, après la phase de testing informel. Se posera alors la question de leur intégration dans les artefacts développés et de leur validation : écriture des cas de tests fonctionnels, remise en question des éléments techniques déjà développés, sous risque de régression,…

On le comprend ici : des requis exprimés partiellement au début d'un projet de cette envergure peuvent avoir des répercussions bien des mois après, et remettre potentiellement en question leur propre validation.

De plus, le projet est susceptible de prendre un retard conséquent, du fait de ce rajout de requis non prévu à l'origine.

- Un travail plus conséquent que prévu

Au-delà des requis ajoutés tardivement, j'ai noté que l'ensemble des travaux du projet a requis plus de temps qu'estimé. Que ce soit les tests unitaires du code d'interface, de la migration et l'écriture des tests, ou du développement de modules spécifiques, le temps prévu a été très souvent dépassé.

Mon analyse m'amène à penser que les raisons viennent de trois facteurs :

- une grande partie des travaux n'a pas connu de précédent : la structure des données venant d'EuroLIMS est en partie unique, dans la mesure où l'application a été en partie développée par Lilly.

De plus, l'application est utilisée par différents sites. Il a donc fallu à l'équipe projet développer l'ensemble des artefacts en tenant des six instances de production, en ne pouvant dans les cas les plus complexes compter que sur des ressources expertes peu disponibles.

- le changement de ressources, qui a impliqué à chaque fois une montée en compétence des nouveaux arrivés, et donc une baisse momentanée du rythme de production. De plus, l'intégration de ces ressources a nécessité une mobilisation des ressources projet déjà en place.

- enfin, le niveau inégal des ressources offshore. Les analystes changeant au sein de la même équipe au début du projet, la démotivation de certains d'entre eux à la fin de la période de contrat ont amené, selon mon analyse, un rythme et une qualité de travail inégaux dans le temps.

- **Bilan professionnel**

L'expérience professionnelle acquise tout au long de ma carrière, mais aussi la formation suivie au CNAM et les conseils reçus m'ont permis de pouvoir prendre en charge ce projet avec des connaissances utiles.

Cette formation mise en pratique au travers de cette expérience de gestion de projet m'a permis de tirer un certain nombre d'enseignements : au-delà de l'aspect technique, qui constitue la base de mon métier, c'est l'acquisition d'un ensemble de compétences non-techniques (gestion d'équipe, gestion de projet, anticipation des risques, coordination avec les

autres équipes informatiques, …) mais aussi un état d'esprit, qui pourront me permettre d'aborder des fonctions différentes dans mon futur professionnel d'ingénieur.

Si je suis aujourd'hui en charge de la coordination d'activités techniques, tant projet que support, il est probable que je sois amené dans le futur à assumer d'autres responsabilités.

Les enseignements tirés de ce projet, et aussi du contexte dans lequel il se déroule, me permettront alors d'appréhender les tâches que d'autres collaborateurs effectueront.

J'ai ainsi retenu les points suivants :

- Des rythmes de travail différents

Les façons de travailler en mode projet et en mode support ne sont pas les mêmes : elles ne s'effectuent pas au même rythme, et ne requièrent pas le même type, ni le même temps de réflexion.

Le support implique une grande réactivité, une répartition de tâches immédiates ou à court terme, une coordination avec d'autres analystes et avec les utilisateurs référents. Les procédures de support et l'expérience de chacun sont mises en œuvre pour résoudre des problèmes urgents. Il y a peu de place pour la recherche de nouvelles solutions informatiques.

Le projet, quant lui, impose un rythme plus lent et plus continu, et demande une réflexion plus large. Il est logique de faire de la recherche et développement, contrairement au support. On privilégie beaucoup plus la communication à deux ou maximum trois personnes, dans la recherche de solutions techniques.

Passer d'un mode à l'autre implique de savoir comprendre rapidement le contexte et d'adopter le rythme de ses collègues, sans quoi on risque de ne pas être assez réactif, ou au contraire des incompréhensions et des erreurs d'analyse, dû à une trop grande rapidité.

De plus, être chef de projet implique non pas de devenir expert technique, contrairement au support, mais de laisser cette expertise se développer chez les analystes, afin de pouvoir prendre en charge plus aisément la direction du projet.

- Préférer une équipe proche

Durant ce projet, mais aussi dans le cadre de ma mission de coordinateur d'équipe support, j'ai confirmé une idée majeure des méthodes de projet agiles : le développement en équipe se révèle être d'autant plus efficient que les analystes travaillent côte à côte, ceci facilitant les échanges et les contributions de chacun.

Les équipes réparties, dites 'virtuelles', se révèlent moins efficaces du fait d'une communication plus difficile, quand bien même elles utilisent des moyens techniques modernes (conférences téléphoniques, messagerie instantanée, sites collaboratifs,…) : éloignement géographique, mais aussi barrières linguistiques et culturelles.

Afin de compenser ces difficultés, il est nécessaire de mettre en place un ensemble de mesures afin d'assurer une bonne communication et une bonne compréhension commune des tâches : points réguliers, partage écrit de toute information ou instruction, mise en place de revues des actions par une deuxième personne.

Ces difficultés se réduisent dans le cas où les membres de l'équipe projet partagent le même espace de travail : par la force des choses, une équipe se créé lorsque ses membres se côtoient.

- L'absence de contrainte peut démotiver

Comme indiqué dans ces pages, le projet n'a pas été soumis à une contrainte de temps très forte, pendant plusieurs mois. De prime abord, ce type de contrainte n'a pas l'air handicapant pour un projet.

Cependant, je me suis rendu compte qu'une date de fin de projet mouvante peut impliquer une démotivation de l'équipe projet, notamment les membres éloignés géographiquement.

Pour y pallier, il m'a fallu régulièrement communiquer avec les membres de l'équipe offshore et m'assurer de leur implication par des revues régulières de leur travail. Les deux analystes présents à mes côtés ont pu plus facilement se rendre compte des conditions de

réalisation du projet et du contexte général, et donc comprendre pourquoi leurs travaux n'obtenaient pas immédiatement de retour de ma part ou des experts Lilly.

De mon côté, je me suis attaché à maintenir un rythme de travail soutenu, en particulier pour les deux ressources basées à Fegersheim. D'une part, pour ne pas perdre leur motivation en attendant une revue ou une information sans autre tâche à effectuer ; d'autre part pour tirer au maximum profit de leur proximité et de leur expertise, afin de faire avancer les différentes activités du projet

- Savoir réagir et savoir déléguer

Un projet est par essence constitué de plusieurs étapes, gérées par une équipe de différentes personnes. Le chef de projet ne saurait participer à la réalisation de chacune d'entre elles. De plus, un certain d'évènements, externes ou internes au projet peuvent survenir et remettre en cause le déroulé du plan de projet.

Le rôle du chef de projet est à mon sens d'analyser ces événements, d'évaluer leur impact sur le projet et d'éventuellement les intégrer dans le plan de projet.

Autrement exprimé : il n'y pas de place pour un contrôle absolu sur le déroulé du projet, à la différence des actions de maintenance par exemple.

La capacité de délégation relève d'une réflexion similaire : on ne peut contrôler aisément la manière dont les analystes travaillent et développent les artefacts. D'une part, cela demanderait d'allouer du temps supplémentaire à cette tâche de contrôle et d'autre part, cela ne permettrait pas aux personnes de monter en compétence et de travailler en confiance. Il est donc essentiel pour le chef de projet de savoir déléguer la réalisation de tâches qu'il pourrait réaliser par lui-même et de laisser de l'autonomie aux développeurs, afin qu'ils puissent développer leur propre expertise.

- Limiter la collecte et organiser l'échange d'informations

Il est important lors d'un projet informatique de capitaliser et de partager les leçons et enseignements tirés par chaque membre de l'équipe afin d'éviter de devoir refaire les mêmes recherches techniques. La démarche d'analyse préalable doit être systématisée avant toute

décision importante. J'ai encouragé cette démarche tout au long du projet, afin de tirer avantage des recherches de chacun. Cependant, j'ai également fait le constat que la collecte systématique d'information (avancée précise de chaque tâche, mesures régulières des temps de transfert ou des volumes de données transférées,…) pouvait créer un trop grand volume d'informations à échanger et à analyser. Une collecte d'informations excessive peut devenir contre-productive et sa diffusion inappropriée grossir inutilement les échanges entre les membres de l'équipe projet.

Il convient donc de limiter la collecte d'informations techniques aux points déterminés comme cruciaux par le chef de projet et les analystes techniques. Les informations pertinentes doivent être diffusées ou mises à disposition au moment approprié, aux personnes qui en ont le besoin.

- **Le mot de la fin**

Ce mémoire s'achève donc sur ces lignes, qui présentent la première partie du projet BOLDER R3.1, et du travail que j'y ai accompli.

Celui-ci va entamer la phase de testing informel et la suite des activités devraient me permettre de mettre à profit les enseignements tirés, mais m'amèneront aussi à faire face à des situations nouvelles, et à prendre des décisions de manière à faire avancer le projet dans le respect des contraintes imposées.

Le projet BOLDER R3.1 est un projet conséquent, en termes de volumes de données, d'artefacts développés ou écrits, d'actions de validation.

Le contexte dans lequel il se déroule ne facilite pas son avancée : les changements de ressources, l'ajout de requis après la première phase de testing, le renouvellement de la contrainte temps sont autant de paramètres dont je devrai tenir compte dans les prochains mois.

L'expérience professionnelle acquise au final comptera sûrement beaucoup dans ma carrière professionnelle.

Table des annexes

- Annexe 1 - Extrait n°1 du cahier des charges BOLDER R3.1 ..... p114
- Annexe 2 - Extrait n°2 du cahier des charges BOLDER R3.1 ..... p115
- Annexe 3 -Méthode de gestion de projet en vigueur au sein du Global MI&CS Lab p116
- Annexe 4 - Déroulé temporel des actions du projet p117
- Annexe 5 - Extrait du plan de projet p118
- Annexe 6 - Extrait du document de design de la base de données : table TEST p119
- Annexe 7 - Extrait des normes de codage pour le langage Ada!Rec p120
- Annexe 8 - Extrait du script Ada!Rec ‘adapter’ pour la table des marchés p121

Annexe 1 - Extrait n°1 du cahier des charges BOLDER R3.1

Number	Definition	Originator	Reference	Priority	Result
RQ.TRN.93	SUPPLIER.DESCRIPTION SUPPLIER.MODIFIED_BY SUPPLIER.MODIFIED_ON SUPPLIER.REMOVEFLAG SUPPLIER.SUPPLIER_ID SUPPLIER.SUPPLIER_STATUS_ID	EuroLIMS retirement project	CR06136768	Critical	Release 3.1
RQ.TRN.106	For all the fields from RQ.TRN.27 to RQ.TRN.105 and RQ.TRN.114, any insertion of a record in EuroLIMS related table should be reflected in the BOLDER database after the next data transfer	EuroLIMS retirement project	CR06136768	Critical	Release 3.1
RQ.TRN.107	For all the fields from RQ.TRN.27 to RQ.TRN.105 and RQ.TRN.114 , any modification of a record in EuroLIMS related table (if this action exists for the table) should be reflected in the BOLDER database after the next data transfer	EuroLIMS retirement project	CR06136768	Critical	Release 3.1
RQ.TRN.108	For all the fields from RQ.TRN.27 to RQ.TRN.105 and RQ.TRN.114, any deletion of a record in EuroLIMS related table (if this action exists for this table) should be reflected in the BOLDER database after the next data transfer	EuroLIMS retirement project	CR06136768	Critical	Release 3.1
RQ.TRN.109	For all the fields from RQ.TRN.27 to RQ.TRN.105 and RQ.TRN.114, a committed record in EuroLIMS related table (if this action exists for this table) should be reflected in the BOLDER database after the next data transfer	EuroLIMS retirement project	CR06136768	Critical	Release 3.1

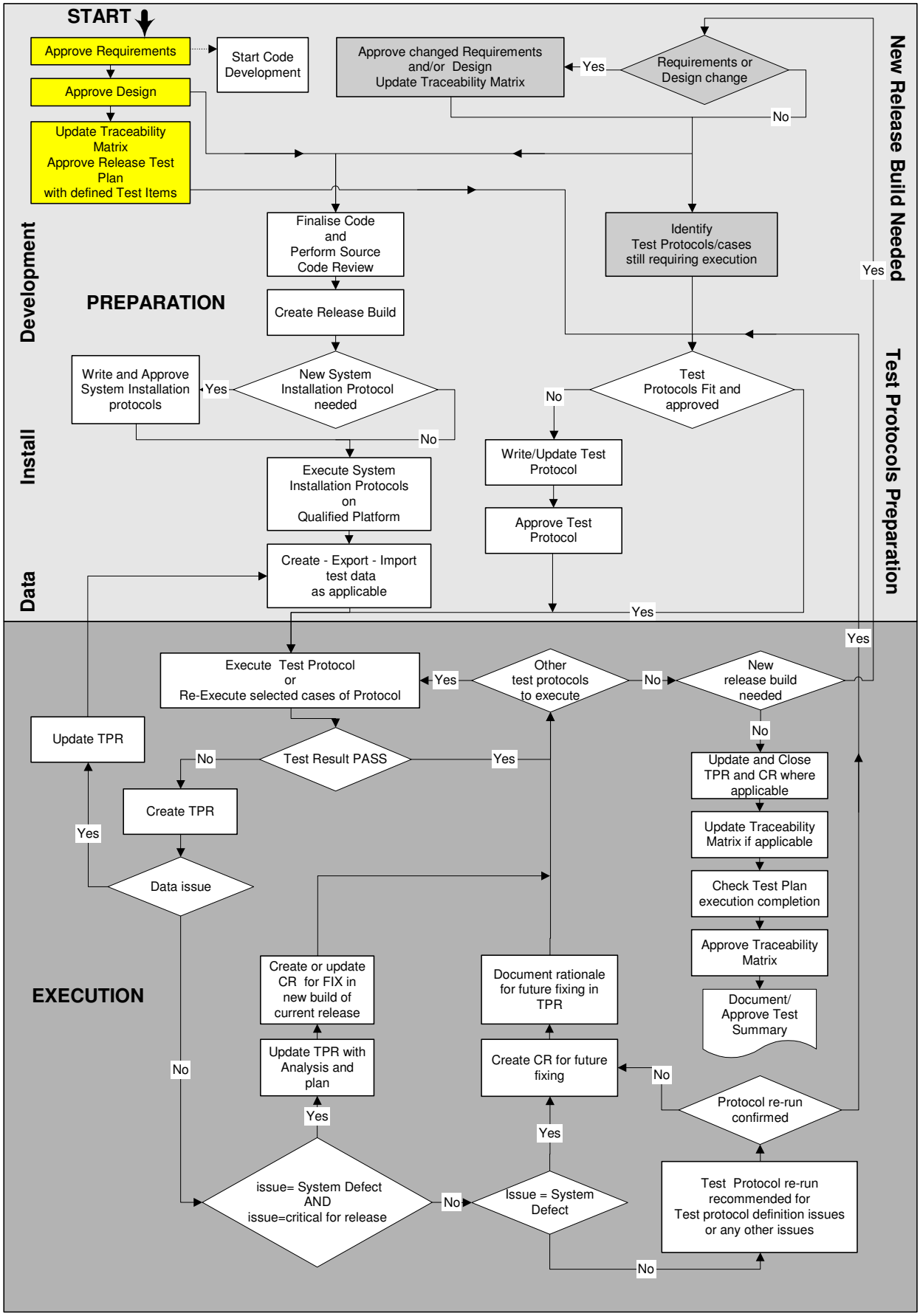
Représentation des données EuroLIMS requises pour le transfert dans BOLDER.

Annexe 2 - Extrait n°2 du cahier des charges BOLDER R3.1

Number	Definition	Originator	Reference	Priority	Result
RQ.SYS.1	BOLDER is a tool that needs to be able to extract EuroLIMS data	Original Project Team	CE16958	Critical	Release 2
RQ.SYS.4.1	No request must last more than 20 minutes	Original Project Team Fegersheim	Release 1 CE04043421	Important	Release 1 Release 2.1
RQ.SYS.4.2	If, after 20 minutes, a request has not completed execution, it must stop and return a partial result.	Original Project Team Fegersheim	Release 1 CE04043421	Important	Release 1 Release 2.1
RQ.SYS.12	BOLDER reports can be generated and/or refreshed using InfoView	Original Project Team	CE04109193	Critical	Release 3.0
RQ.SYS.13	BOLDER reports can be generated and/or refreshed using Desktop Intelligence	Original Project Team	N/A	Critical	Release 3.1
RQ.SYS.14	BOLDER must be validated to use Oracle 11g as Oracle database version.	Original Project Team	CR06179864	Critical	Release 3.1
RQ.SYS.15	The application must be based on the version XI R3 of Business Objects	Original Project Team	CR06137736	Critical	Release 3.1

Représentation des requis pour la partie système de l'application BOLDER

Annexe 3 - Méthode de gestion de projet en vigueur au sein du Global MI&CS Lab

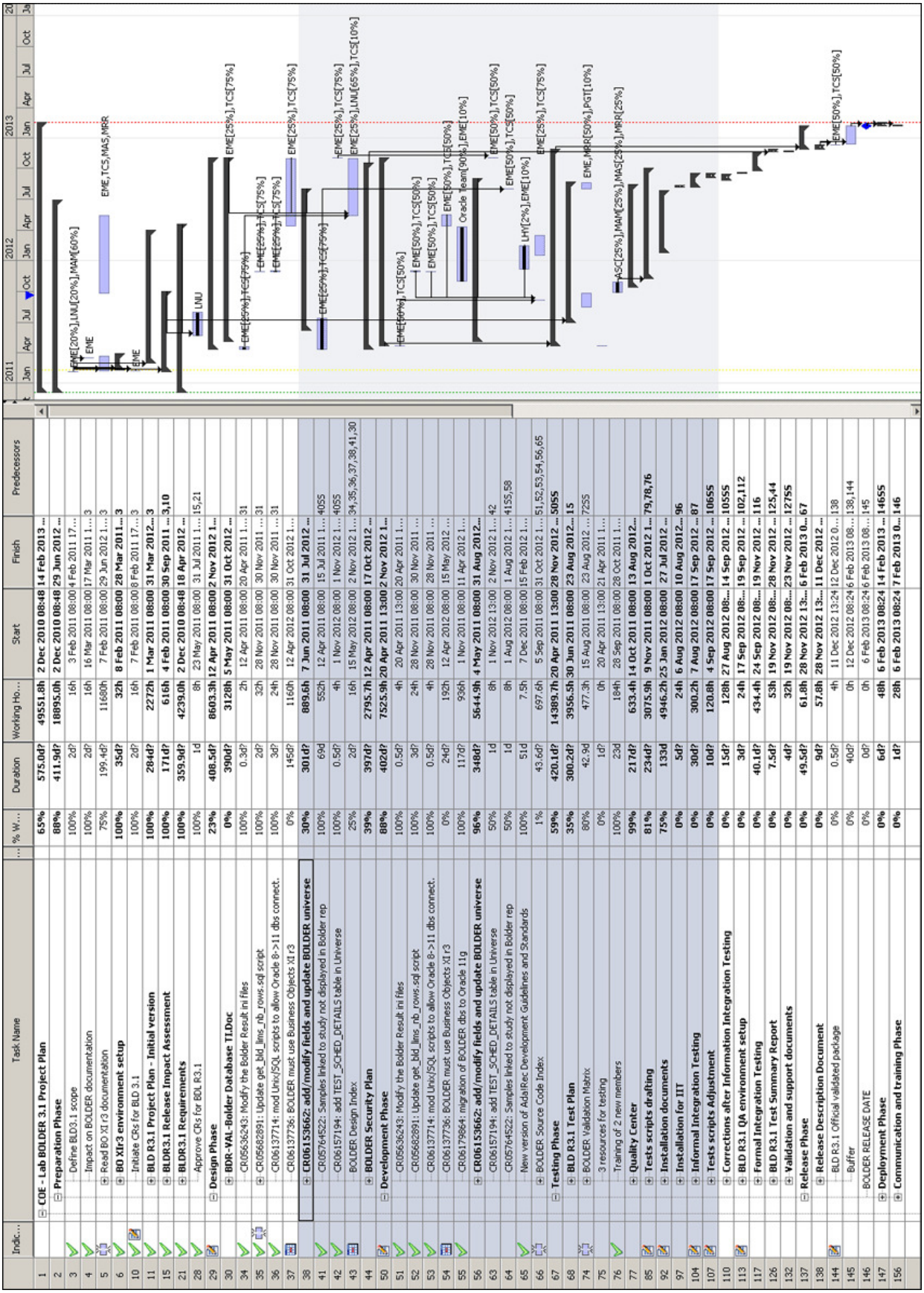


Annexe 4 - Déroulé temporel des actions du projet

Octobre 2010	Février 2011	Mai	Juin	Septembre	Octobre 2011
Tests de compatibilité Ada!Re c/ Oracle 11					
	Réunion de lancement du projet BOLDER R3.1				
		Modélisation de la base de données			
				Renforcement de l'équipe projet	
			Début des travaux de développement de l'interface et de la base de données		
					Début de la migration des tests dans Quality Center

Novembre	Décembre 2011	Mars 2012	Avril	Juin	Août
Suite et fin des travaux de développement de l'interface et de la base de données					
Fin de la migration des tests dans Quality Center					
Travaux de développement pour le transfert des fichiers et des BLOB					
		Formations données aux nouveaux analystes offshore (hors projet)			
			Première phase de travaux de développement de l'univers Business Objects		
					Début du testing informel

Annexe 5 - Extrait du plan de projet



Indic...	Task Name	...	% W...	Duration	Working Ho...	Start	Finish	Predecessors
1	CDE - Lab BOLDER 3.1 Project Plan		65%	575.0d?	49551.8h	2 Dec 2010 08:48	14 Feb 2013 ...	
2	Preparation Phase		88%	411.9d?	18895.0h	2 Dec 2010 08:48	29 Jun 2012 ...	
3	Define BLD3.1 scope		100%	2d?	16h	3 Feb 2011 08:00	4 Feb 2011 17:...	
4	Impact on BOLDER documentation		100%	2d?	16h	16 Mar 2011 08:00	17 Mar 2011 1:...	3
5	Read BO XI r3 documentation		75%	199.4d?	11680h	7 Feb 2011 08:00	29 Jun 2012 1:...	3
6	BO XIr3 environment setup		100%	35d?	32h	8 Feb 2011 08:00	28 Mar 2011 ...	3
10	Initiate CRs for BLD 3.1		100%	2d?	16h	7 Feb 2011 08:00	8 Feb 2011 17:...	3
11	BLD R3.1 Project Plan - Initial version		100%	28d?	2272h	1 Mar 2011 08:00	31 Mar 2012...	3
15	BLDR3.1 Release Impact Assessment		100%	171d?	616h	4 Feb 2011 08:00	30 Sep 2011 ...	3,10
21	BLDR3.1 Requirements		100%	359.9d?	4239.0h	2 Dec 2010 08:48	18 Apr 2012 ...	
28	Approve CRs for BDL R3.1		100%	1d	8h	23 May 2011 08:00	31 Jul 2011 1:...	15,21
29	Design Phase		23%	408.5d?	8603.3h	12 Apr 2011 08:00	2 Nov 2012 1...	
30	BDR-VAL-Bolder Database TI.Doc		0%	390d?	3128h	5 May 2011 08:00	31 Oct 2012 ...	
34	CR05636243: Modify the Bolder Result ini files		100%	0.3d?	2h	12 Apr 2011 08:00	20 Apr 2011 1:...	31
35	CR05682891: Update get_bld_lims_rb_rows.sql script		100%	2d?	32h	28 Nov 2011 08:00	30 Nov 2011 ...	31
36	CR06137714: mod Unix/SQL scripts to allow Oracle 8->11 dbs connect.		100%	3d?	24h	28 Nov 2011 08:00	30 Nov 2011 ...	31
37	CR06137736: BOLDER must use Business Objects XI r3		0%	145d?	1160h	12 Apr 2012 08:00	31 Oct 2012 1:...	
38	CR06153662: add/modify fields and update BOLDER universe		30%	301d?	889.6h	7 Jun 2011 08:00	31 Jul 2012 ...	
41	CR05764522: Samples linked to study not displayed in Bolder rep		100%	69d	552h	12 Apr 2011 08:00	15 Jul 2011 1:...	4055
42	CR06157194: add TEST_SCHED_DETAILS table in Universe		100%	0.5d?	4h	1 Nov 2012 08:00	1 Nov 2012 1:...	4055
43	BOLDER Design Index		25%	2d?	16h	15 May 2012 08:00	2 Nov 2012 1:...	34,35,36,37,38,41,30
44	BOLDER Security Plan		39%	397d?	2795.7h	12 Apr 2011 08:00	17 Oct 2012 ...	
50	Development Phase		88%	402d?	7525.9h	20 Apr 2011 13:00	2 Nov 2012 1...	
51	CR05636243: Modify the Bolder Result ini files		100%	0.5d?	4h	20 Apr 2011 13:00	20 Apr 2011 1:...	
52	CR05682891: Update get_bld_lims_rb_rows.sql script		100%	3d?	24h	28 Nov 2011 08:00	30 Nov 2011 ...	
53	CR06137714: mod Unix/SQL scripts to allow Oracle 8->11 dbs connect.		100%	0.5d?	4h	28 Nov 2011 08:00	28 Nov 2011 ...	
54	CR06137736: BOLDER must use Business Objects XI r3		0%	24d?	192h	12 Apr 2012 08:00	15 May 2012 ...	
55	CR06179864: migration of BOLDER dbs to Oracle 11g		100%	117d?	936h	1 Nov 2011 08:00	11 Apr 2012 1:...	
56	CR06153662: add/modify fields and update BOLDER universe		96%	348d?	5644.9h	4 May 2011 08:00	31 Aug 2012...	
63	CR06157194: add TEST_SCHED_DETAILS table in Universe		50%	1d	8h	1 Nov 2012 13:00	2 Nov 2012 1:...	42
64	CR05764522: Samples linked to study not displayed in Bolder rep		50%	1d	8h	1 Aug 2012 08:00	1 Aug 2012 1:...	4155,58
65	New version of AdaRec Development Guidelines and Standards		100%	51d	7.5h	7 Dec 2011 08:00	15 Feb 2012 1:...	
66	BOLDER Source Code Index		1%	43.6d?	697.6h	5 Sep 2011 08:00	31 Oct 2012 1:...	51,52,53,54,56,65
67	Testing Phase		59%	420.1d?	14389.7h	20 Apr 2011 13:00	28 Nov 2012 ...	5055
68	BLD R3.1 Test Plan		35%	300.2d?	3956.5h	30 Jun 2011 08:00	23 Aug 2012 ...	15
74	BOLDER Validation Matrix		80%	42.9d	477.3h	15 Aug 2011 08:00	23 Aug 2012 ...	7255
75	3 resources for testing		0%	1d?	0h	20 Apr 2011 13:00	21 Apr 2011 1:...	
76	Training of 2 new members		100%	23d	184h	28 Sep 2011 08:00	28 Oct 2011 1:...	
77	Quality Center		99%	217d?	633.4h	14 Oct 2011 08:00	13 Aug 2012...	
85	Tests scripts drafting		81%	234d?	3075.9h	9 Nov 2011 08:00	1 Oct 2012 1...	79,78,76
92	Installation documents		75%	133d	4946.2h	25 Jan 2012 08:00	27 Jul 2012 ...	
97	Installation for IIT		0%	5d?	24h	6 Aug 2012 08:00	10 Aug 2012...	96
104	Informal Integration Testing		0%	30d?	300.2h	7 Aug 2012 08:00	17 Sep 2012 ...	87
107	Tests scripts Adjustment		0%	10d?	120.8h	4 Sep 2012 08:00	17 Sep 2012 ...	10655
110	Corrections after Information Integration Testing		0%	15d?	128h	27 Aug 2012 08:...	14 Sep 2012 ...	10555
113	BLD R3.1 QA environment setup		0%	3d?	24h	17 Sep 2012 08:...	19 Sep 2012 ...	102,112
117	Formal Integration Testing		0%	40.1d?	434.4h	24 Sep 2012 08:...	19 Nov 2012 ...	116
126	BLD R3.1 Test Summary Report		0%	7.5d?	53h	19 Nov 2012 08:...	28 Nov 2012 ...	125,44
132	Validation and support documents		0%	4d?	32h	19 Nov 2012 08:...	23 Nov 2012 ...	12755
137	Release Phase		0%	49.5d?	61.8h	28 Nov 2012 13:...	6 Feb 2013 0...	67
138	Release Description Document		0%	9d?	57.8h	28 Nov 2012 13:...	11 Dec 2012 ...	
144	BLD R3.1 Official validated package		0%	0.5d?	4h	11 Dec 2012 13:24	12 Dec 2012 0:...	138
145	Buffer		0%	40d?	0h	12 Dec 2012 08:24	6 Feb 2013 08:...	138,144
146	BOLDER RELEASE DATE		0%	0d?	0h	6 Feb 2013 08:24	6 Feb 2013 08:...	145
147	Deployment Phase		0%	6d?	48h	6 Feb 2013 08:24	14 Feb 2013 ...	14655
156	Communication and training Phase		0%	1d?	28h	6 Feb 2013 08:24	7 Feb 2013 0...	146

Annexe 6 - Extrait du document de design de la base de données : table TEST

TEST				
Field name	Null	Attribute	Data Definition	Original EuroLIMS Field
TEST_ID	Not Null	PK	INTEGER	TEST.TEST_NUMBER
ANALYSIS_ID	Not Null	FK1 I1	VARCHAR2(10)	TEST.ANALYSIS
ANALYSIS_VERSION	Not Null	FK1	VARCHAR2(10)	(1°)
SAMPLE_ID_NUM	Not Null	FK2 I2	INTEGER	TEST.SAMPLE
STATUS	Not Null		VARCHAR2(1)	TEST.STATUS
STARTED_ON			DATE	TEST.DATE_STARTED
STARTED_BY			VARCHAR2(10)	TEST.STARTER
COMPLETED_ON		I3	DATE	TEST.DATE_COMPLETED
AUTHORISED_ON		I4	DATE	TEST.DATE_AUTHORISED
AUTHORISED_BY			VARCHAR2(10)	TEST.AUTHORISER
AUTORISATION_COMMENT			VARCHAR2(234)	TEST.AUTORISATION_COMMENT
LAST_TEST_ON			DATE	TEST.LAST_TEST_DATE
ORDER_NUM	Not Null		INTEGER	TEST.ORDER_NUM
PARENT_TEST			VARCHAR2(10)	TEST.PARENT_TEST
INSPECTION_HEADER			VARCHAR2(10)	TEST.INSPECTION_HEADER
VALIDATION_STATUS			VARCHAR2(1)	TEST.VALIDATION_STATUS
AUTO_VALIDATE			VARCHAR2(1)	TEST.AUTO_VALIDATE
HAS_INCIDENTS			VARCHAR2(1)	TEST.HAS_INCIDENTS
STAGE_ID			VARCHAR2(10)	TEST.STAGE_ID
STAGE_DESC			VARCHAR2(234)	STAGE.DESCRIPTION
STAGE_MODIFIED_ON			DATE	STAGE.MODIFIED_ON
STAGE_MODIFIED_BY			VARCHAR2(10)	STAGE.MODIFIED_BY
LOT_ID		I5	VARCHAR2(10)	TEST.LOT_ID
PREPARATION			VARCHAR2(10)	TEST.PREPARATION
PREPARATION_DESC			VARCHAR2(234)	PREPARATION.DESCRIPTION
PREPARATION_MODIFIED_ON			DATE	PREPARATION.MODIFIED_ON
PREPARATION_MODIFIED_BY			VARCHAR2(10)	PREPARATION.MODIFIED_BY
TEST_COUNT			INTEGER	TEST.TEST_COUNT
COMPLETER			VARCHAR2(10)	TEST.COMPLETER
REPLICATE_TEST			VARCHAR2(1)	TEST.REPLICATE_TEST
ON_SPEC			VARCHAR2(1)	TEST.ON_SPEC
TEST_SCHEDULE			VARCHAR2(20)	TEST.TEST_SCHEDULE
TEST_SCHED_DESC			VARCHAR2(234)	TEST_SCHED_HEADER.DESCRIPTION
TEST_SCHED_MODIFIABLE			VARCHAR2(1)	TEST_SCHED_HEADER.MODIFIABLE
TEST_SCHED_MODIFIED_ON			DATE	TEST_SCHED_HEADER.MODIFIED_ON
TEST_SCHED_MODIFIED_BY			VARCHAR2(10)	TEST_SCHED_HEADER.MODIFIED_BY
TEST_SCHED_GROUP_ID			VARCHAR2(10)	TEST_SCHED_HEADER.GROUP_ID
COMPONENT_LIST			VARCHAR2(10)	TEST.COMPONENT_LIST
STAT_DOUBLE_ATT_FAIL			VARCHAR2(1)	TEST.STAT_DOUBLE_ATT_FAIL
NOTEBOOK			VARCHAR2(20)	TEST.NOTEBOOK
POPULATED_DATE	Not Null	I6	DATE	INTERNAL
MAPPING_KEY0			INTEGER	INTERNAL

Annexe 7 - Extrait des normes de codage pour le langage Ada!Rec

Part 2 : Code Review Acceptance Criteria

The Reviewer is responsible for checking that the code conforms to the following checklist (for the definition of standards refer to the *Ada!Rec – Development Standards & Guidelines)*.

Yes	No	N/A	Acceptance Criterion
			Adherence to coding/programming standards
			Structure
❒	❒	❒	The INI file starts with a Header in which the following information appears : Filename, Author, Version, Short description as a comment
❒	❒	❒	If applicable, the Header is completed with Revision history, with blocks in which the following information appears : Modifier, Date, Version, Short description of changes and its number as a comment
❒	❒	❒	If the INI file configures an Adapter, the following clauses are defined : SQLCON, SQLGET, DMTOOL and at least one FIELD
❒	❒	❒	If the INI file configures a Receiver, the following clauses are defined : SQLCON, at least one SQLCMD, DMTOOL and at least one FIELD
❒	❒	❒	The clauses in the INI file respect the sequence order defined in the templates
			Appearance
❒	❒	❒	The clauses and environment variables specific to the Adapter or receiver appear in uppercase characters
❒	❒	❒	SQL blocks defined using ENDSQL keyword are indented
❒	❒	❒	When used, ENDSQL is placed on the same column than its corresponding SQL clause
❒	❒	❒	No redefinition of non-repetitive clause appear in INI file
❒	❒	❒	Comments are written in English, are useful and sufficient
❒	❒	❒	Oracle keywords are in uppercase character, and table/field names are lowercase
			Naming Conventions
❒	❒	❒	The fields defined by the FIELD clause have their names respecting the standard format [*ConditionCode_*]*DatabaseFieldName*
			Security
❒	❒	❒	If the version of Ada!Rec used is 3.1.0 or above, the Oracle password is crypted in the SQLCON clause
			Critical parameters incorporated
❒	❒	❒	Source Code has been checked for accuracy against the specifications
			Dead Code
❒	❒	❒	No commented out code appear in the INI file

Annexe 8 - Extrait du script Ada!Rec 'adapter' pour la table des marchés.

```
[…]
SQLMRK
UPDATE bld_itf_market
     SET row_status='L'
   WHERE row_status='O'
     AND row_closed_on IS NULL
ENDSQL
#
# Clause to extract data
SQLGET
SELECT mk.market_id,
            mk.description,
            -- Begin addition for CR06153662 --
            decode( to_char( mk.modified_on, 'YYYYMMDD' ),
            '00000000',
            NULL,
            to_char( mk.modified_on, 'YYYYMMDD HH24:MI:SS' ) )
            modified_on,
            mk.modified_by,
            -- End addition for CR06153662 --
            mk.removeflag
FROM market mk
WHERE mk.market_id IN
            ( SELECT blditf.market_id
             FROM bld_itf_market blditf
            WHERE blditf.row_status='L'
            AND blditf.row_closed_on IS NULL )
ENDSQL
#
[…]
# Clause to mark data as transferred
SQLUPD
UPDATE bld_itf_market
  SET row_status='C',
      row_closed_on = SYSDATE
 WHERE row_status='L'
  AND row_closed_on IS NULL
ENDSQL
#
# Clause to clean up old data transferred
SQLHSK
DELETE FROM bld_itf_market
 WHERE row_status = 'C'
  and row_closed_on < ( SYSDATE - $(BLD_HOUSEKEEPING_TIME) )
ENDSQL
[…]
#
# description of fields : <field_name>,<field_db_length>[,field_AOF_length]
FIELD=mk.market_id,10
FIELD=mk.description,60
# Begin addition for CR06153662
FIELD=modified_on,17
FIELD=mk.modified_by,10
```

Liste des figures

Figure 1 - Carte des implantations des établissements Lilly dans le monde p14
Figure 2 - Structure schématique de l'équipe support EuroLIMS/BOLDER p15
Figure 3 - Schéma des processus des métiers de laboratoires, intervenant lors de la fabrication d'un lot de produit p18
Figure 4 - Fonctionnement de base des tests dans EuroLIMS p21
Figure 5 - Système de comparaison des résultats aux limites p22
Figure 6 - Schéma expliquant la fonction des Quality Masters p23
Figure 7 - Schéma de l'approbation d'un lot de produit p24
Figure 8 - Schéma d'aperçu de l'infrastructure de BOLDER p27
Figure 9 - Schéma explicatif d'un transfert de données incrémental : table des marchés p28
Figure 10 - Méthode de projet du Global MI&CS Lab : étapes initiales du projet p47
Figure 11 - Méthode de projet du Global MI&CS Lab : installation et écriture des tests p47
Figure 12 - Méthode de projet du Global MI&CS Lab : étapes à respecter lors des tests p48
Figure 13 - Méthode de projet du Global MI&CS Lab : cas d'un test échoué pour cause de données p49
Figure 14 - Méthode de projet du Global MI&CS Lab : nouvelle itération de tests p50
Figure 15 - Schéma équipe projet des débuts p55
Figure 16 - Schéma équipe projet renforcée p56
Figure 17 - Extrait du plan du projet BOLDER R3.1 : vue générale simple p60
Figure 18 - Extrait du plan du projet BOLDER R3.1 : vue détaillée de réserve de temps p60
Figure 19 - Modèle en étoile pour la table TEST dans EuroLIMS p68
Figure 20 - Schéma du modèle dénormalisé pour la table TEST dans BOLDER p69
Figure 21 - Extrait de la série de tests BOLDER dans Quality Center : tests IT.TR N17.x p73
Figure 22 - Extrait de la matrice de traçabilité, pour le transfert de la table 'Product lot'. p76
Figure 23 - Extrait de la matrice de traçabilité, onglet montrant la correspondance entre anciens et nouveaux tests p80
Figure 24 - Processus de base de gestion des fichiers de code p78
Figure 25 - Schéma du processus de gestion des fichiers de code – projet BOLDER R3.1 ..... p82
Figure 26 - Extrait de l'application EuroLIMS : fichier de code utilisé pour calculer des résultats complexes p89
Figure 27 - Extrait de l'application EuroLIMS : données stockées en BLOB, présentées en impression virtuelle p89
Figure 28 - Extrait du tableau d'estimation des temps d'exécution par test p101

Liste des tableaux

Tableau 1 - Répartition des offres de service de Prodaxis p11
Tableau 2 - Répartition des activités de tests laborantins dans les processus métiers p17
Tableau 3 - Récapitulatif des données de base d'EuroLIMS par instance de production p26
Tableau 4 - Récapitulatif des requis du cahier des charges, classés par type p41
Tableau 5 - Récapitulatif des responsabilités pour le projet BOLDER R3.1 ..... p54
Tableau 6 - Liste des phases du plan de projet p59
Tableau 7 - Récapitulatif des fichiers des versions BOLDER R3.0 et R3.1 ..... p77
Tableau 8 - Récapitulatif des volumes à transférer – instances au mois de Juillet 2012 ..... p87
Tableau 9 - Récapitulatif des fichiers et données de BLOB à transférer – instances au mois de Juillet 2012 ..... p90
Tableau 10 - Récapitulatif des ajouts d'artefacts dans l'univers Business Objects p95
Tableau 11 - Récapitulatif des totaux de tests par types p99

Reporting et archivage de données d'un SIGL

Mémoire d'Ingénieur C.N.A.M., Strasbourg 2012

RESUME

Le futur retrait d'un progiciel SIGL utilisé dans une compagnie internationale pharmaceutique internationale, a entraîné la modification de la solution de reporting associée.

Ce mémoire présente les activités de gestion de ce projet, qui se déroule dans un contexte de travail international, où les aspects documentaire et de validation sont capitaux et requièrent une méthode de projet adaptée.

Il présente également les activités et les démarches du chef de projet durant les phases de mise en place, développement et pré-testing, et annonce les futures étapes du projet.

Mots clés : reporting, SIGL, LIMS, pharmaceutique, archivage, retrait, Business Objects, offshore.

SUMMARY

The coming retirement of a LIMS software, being used in an international pharmaceutical compagny, results in the upgrade of the association reporting application.

This essay presents the project's management activities, which are taking place in a international context where the documentary and validation aspects are essential and require an adapted project's framework.

It also presents the activities and the approaches of the project's manager during the phases of setting up, development and pre-testing, and announces the next steps of the project.

Key words : reporting, LIMS, pharmaceutical, archiving, retirement, Business Objects, offshore.