



HAL
open science

Évolution technologique et structurelle en vue d'externaliser les développements internes

Sébastien Salameh

► **To cite this version:**

Sébastien Salameh. Évolution technologique et structurelle en vue d'externaliser les développements internes. Multimédia [cs.MM]. 2012. dumas-01157672

HAL Id: dumas-01157672

<https://dumas.ccsd.cnrs.fr/dumas-01157672>

Submitted on 28 May 2015

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.

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

CENTRE D'AIX-EN-PROVENCE

**ORGANISME DE FORMATION FINANÇÉ PAR LE CONSEIL RÉGIONAL
DE LA RÉGION PROVENCE-ALPES-COTE D'AZUR**

MÉMOIRE

**PRESENTÉ EN VUE D'OBTENIR
LE DIPLÔME D'INGENIEUR C N A M**

en

INFORMATIQUE

Option Réseaux-Systèmes-Multimédia

par

Sébastien SALAMEH

**ÉVOLUTION TECHNOLOGIQUE ET STRUCTURELLE EN VUE
D'EXTERNALISER LES DEVELOPPEMENTS INTERNES**

Soutenu le 13 Juillet 2012

Jury :

Président :

M. Y. LALOUM

Membres :

M. Jean-Louis BRUNEL

M. Erwan TRANVOUEZ

M. Xavier VELUT (milieu professionnel)

M. Romain BOIS (milieu professionnel)

Sébastien SALAMEH – Évolution technologique et structurelle en vue d'externaliser les développements internes

Mémoire d'ingénieur C.N.A.M. PACA 2012

Développer son système d'information et son logiciel de travail collaboratif est un véritable défi pour une PME, en pleine évolution. Au fur et à mesure, j'en ai pris conscience lors de mon intégration dans l'entreprise. Ce qui m'a particulièrement attiré c'est le savoir-faire existant au sein du groupe ainsi que sa flexibilité afin de mener des projets de fond.

Les ambitions portées par la direction couplées à un certain nombre de dysfonctionnements identifiés ont conduit à l'évolution technologique et structurelle présentée dans ce mémoire. De nos différentes réflexions nous avons souvent établi les mêmes conclusions, en revanche, les évolutions nécessaires devaient être accompagnées d'autres objectifs moteurs, assurant un avenir intéressant sur du long terme.

C'est donc dans ce contexte que nous avons pris la décision conjointe de faire évoluer notre système d'information afin de pouvoir recouvrir l'effort d'investissement précédent et externaliser nos productions. Ainsi notre potentiel mis en œuvre en interne depuis des années va donc être ouvert à d'autres structures sur des secteurs d'activités que nous allons cibler.

Alors que de fixer les objectifs est déjà un premier engagement au changement, leur réalisation n'est pas sans conséquence pour une entreprise qui doit encore une fois investir et réorganiser sensiblement son fonctionnement. Nous avons donc prêté une attention particulière à la réalisation d'une évolution mesurée sans risquer de mettre en péril la croissance actuelle du groupe.

Ce dernier point est plus fort qu'un objectif, c'est une ligne directrice qui doit être omniprésente tout au long du projet. En effet, il est essentiel de ne pas sous-estimer les systèmes en production à l'heure actuelle qui sont générateurs du chiffre d'affaire annuel.

Dès mon arrivée dans l'entreprise j'ai pris toutes mes responsabilités dans mes missions, tout en essayant de faire preuve d'initiative et d'innovation. Je pense que c'est pour cette raison que la direction m'a confié la gestion de ce projet dans sa globalité.

Information system and workflow software development is a real challenge for an SME in evolution. I realized this during my company integration. I have been particularly attracted by the know-how and the company flexibility to conduct structural changes.

Leadership ambitions as well as a number of identified dysfunctions have led to structural and technological evolution presented in this thesis. Our different reflections have often led to this conclusion, but these changes should go along with other strong objectives ensuring a positive future over the long term.

Considering this context we decided to develop our Information system in order to cover previous investments and outsource our previous productions. The products we developed so far for our core business will be available for new customers on sectors that we will target.

While setting goals is an imperative requirement to change, achieving these goals is not without consequence for a company that must once again invest and reorganize its operations significantly. We will therefore pay particular attention to achieve this evolution without jeopardizing the current growth of the group.

The latter is stronger than a goal it is a guideline we need to keep in mind all along the project. It is crucial to maintain the systems in production today which produce the annual turnover.

Since I arrived in the company I took my responsibility in my missions and I proved my entrepreneurial spirit and my ability to innovate. I think that led the directors to put me in charge of this project.

Mots-Clés : Évolution technologique et structurelle – Développement de solution logicielle de travail collaboratif

Keywords : Technological and structural evolution – Development of workflow software solution

CONSERVATOIRE NATIONAL DES ARTS ET MÉTIERS

CENTRE D'AIX-EN-PROVENCE

ORGANISME DE FORMATION FINANÇÉ PAR LE CONSEIL RÉGIONAL
DE LA RÉGION PROVENCE-ALPES-COTE D'AZUR

MÉMOIRE

**PRESENTÉ EN VUE D'OBTENIR
LE DIPLÔME D'INGENIEUR C N A M**

en

INFORMATIQUE

Option Réseaux-Systèmes-Multimédia

par

Sébastien SALAMEH

**ÉVOLUTION TECHNOLOGIQUE ET STRUCTURELLE EN VUE
D'EXTERNALISER LES DEVELOPPEMENTS INTERNES**

Soutenu le 13 Juillet 2012

Les travaux relatifs au présent mémoire ont été effectués au sein de la société INSIIDE, sous la direction de Monsieur Frédéric VINCENT.

SOMMAIRE

1.	Introduction.....	5
1.1	Présentation du groupe.....	5
1.2	Le système d'information	9
1.3	Présentation générale du projet	10
1.4	Ligne directrice du projet.....	13
2.	Réorganisation structurelle	16
2.1	Insiide	16
2.2	Group Mayet Consultant	18
2.3	Bilan	20
3.	Réorganisation technique	21
3.1	Fonctionnelle	21
3.2	L'évolution de l'architecture.....	22
3.3	Éléments de planning	25
3.4	Bilan	27
4.	Évolution d'Insiide.....	28
4.1	La gestion de projet	28
4.2	Recherche et développement	29
4.3	Définition des cahiers des charges	29
4.4	Solution retenue.....	32
5.	Réalisation technique.....	35
5.1	Avant le développement	35
5.2	Développement et consolidation	39
5.3	Recette et mise en production	49
6.	Conclusion.....	51
7.	Glossaire	54
8.	Table des illustrations	56
9.	Annexe	57
I.	Couverture fonctionnelle	57
II.	Suivi projet.....	61
III.	Architecture du système d'information	61
IV.	Modèle conceptuel de données (MCD)	62
I.	MCD Nouveau produit	62

1. Introduction

Le contexte économique des entreprises conduit à avoir des réflexions de fond sur la mutualisation des activités afin d'obtenir un retour sur investissement des efforts menés jusqu'à présent.

En effet l'entreprise, plus précisément le groupe, dans lequel j'ai évolué, a toujours fait le choix d'investir pour la création de son propre système d'information. Celui-ci est directement adapté à son métier et cela représente un réel atout. Face à la concurrence, il représente une flexibilité offerte par un logiciel de gestion modulable à la demande. Les dirigeants ont également su saisir des opportunités de développement à destination d'autres métiers que leur activité principale.

Les besoins ont donc progressés avec l'évolution des différentes activités du groupe. À cet effet, le système d'information doit aujourd'hui faire partie d'une restructuration globale pour s'adapter à ces nouvelles exigences.

Group Mayet Consultant, est un groupe spécialisé dans l'expertise et la gestion de sinistre.

1.1 Présentation du groupe

Le groupe « Group Mayet Consultant », [Figure 1](#), est composé de deux structures qui gèrent respectivement les familles d'activités « grand risque », « bien de masse » et « immobilier ». On retrouve ainsi :

Les activités principales

GM Consultant (GMC)

Il s'agit d'un cabinet d'expertise en charge de la famille « Grand Risque ». Cette dernière correspond aux activités dont l'impact des risques entraîne des conséquences importantes du point de vue humain, économique et industriel. GMC travaille essentiellement avec des assureurs. Il est composé d'experts spécialisés par pôle d'activité qui interviennent soit dans le cadre d'une demande établie soit lors d'une délégation complète de dossier par le client.

Société Européenne de Sauvetage (SES)

SES est une société de gestion d'adhésion et de sinistre. La famille « Bien de Masse » couvre des risques de moindre importance et qui touche des produits ou services ne dépassant pas un plafond établi. Elle est amenée à travailler avec un panel beaucoup plus large de clients. Ce sont des assureurs, des gestionnaires de flotte de grande distribution ou encore des marques qui proposent une assurance dont la gestion nous ait déléguée.

SES Immo correspond à une spécialité de gestion des risques immobiliers. La famille « Immobilier » couvre un domaine qui consiste à prendre en charge la gestion de ces sinistres spécifiques isolés du bilan par les entreprises. C'est une structure qui gère le plus souvent des situations de conflits aggravés entre deux tiers. Son expertise consiste à gérer, lorsque cela est possible, celles-ci avec des collaborateurs formés sur les sujets juridiques. En effet, les assureurs que nous représentons souhaitent le moins possible avoir recours à la justice pour solder ces dossiers.

Les activités secondaires

Parmi les opportunités considérées comme des activités secondaires, nous retrouvons le développement de deux solutions. GRClive (logiciel de gestion de relation client) et Global Business Managment (GBM, logiciel de travail collaboratif) qui sont issues des modules spécialement développés pour assurer la gestion de dossier multi-tiers.

Insiide

La société d'ingénierie informatique a un rôle transversal dans la gestion du système d'information de l'intégralité du groupe. Elle a également comme attribution le développement et la maintenance de la suite logicielle tout comme les activités secondaires.

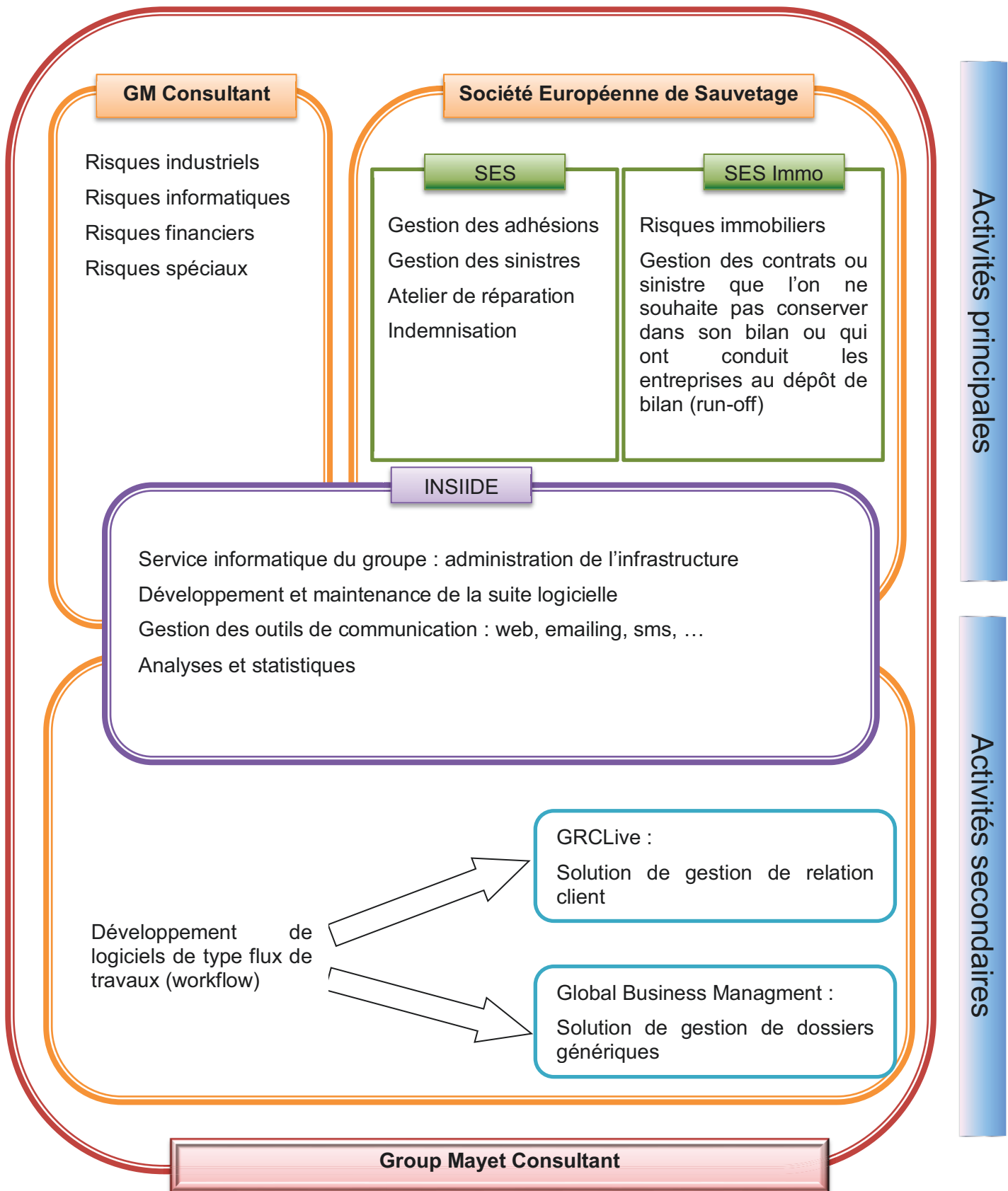


Figure 1 – Représentation des principales activités et de leurs répartitions

La [Figure 1](#) présente trois activités bien distinctes, qui elles même sont encore composées de sous partie. Celles-ci peuvent être fortement différentes en termes de processus de gestion. Cependant, d'un point de vue informatique la gestion de ces sinistres se recoupe sur des fonctions communes.

Aujourd'hui avec plus de 15 ans d'expérience le groupe compte 120 collaborateurs et produit 6,8 Millions de chiffres d'affaire par an. La [Figure 2](#) montre la répartition de celui-ci entre les deux structures historiques. D'un point de vue purement juridique le groupe a scindé son activité en plusieurs entreprises afin de répartir les risques liés à l'activité.

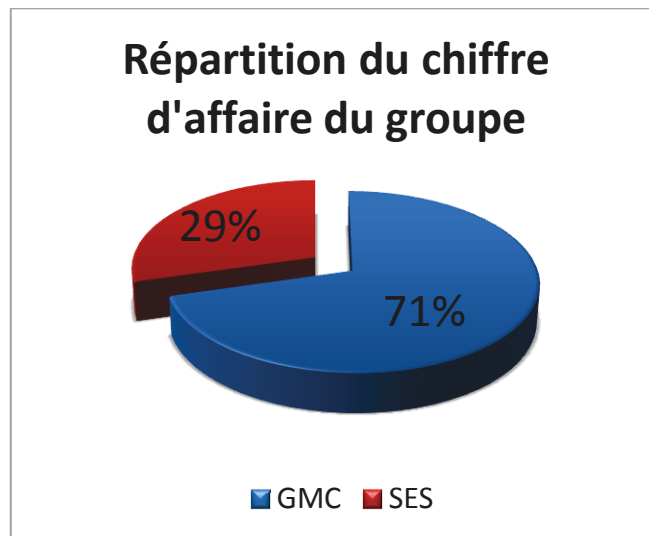


Figure 2 – Répartition du chiffre d'affaire du groupe

Group Mayet Consultant propose historiquement des services additionnels et des prestations toujours plus personnalisées autour de la gestion de sinistre. Ces services autour des technologies de l'information et de la communication sont portés par Insiide. Nous en proposons toute une gamme qui porte comme priorité l'interaction entre les tiers et le dossier.

Les sociétés mères, GMC et SES, ont leurs propres désirs d'évolution et ces derniers ne sont pas nécessairement en phase avec ceux d'Insiide. Durant les huit dernières années le chiffre d'affaire de GMC a été multiplié par quatre, alors que celui de SES a connu une forte augmentation de plus de dix sur la même période. Ces chiffres mettent en évidence la flexibilité du groupe, notamment grâce à son système d'information.

Le métier principal de gestion de sinistre présenté sur la [Figure 3](#) met en évidence :

- les différents acteurs qui gèrent un sinistre et leurs échanges,
- la répartition des tâches de la réception de la mission jusqu'à la clôture du dossier.

Il est important de préciser que j'ai mis en évidence un cas représentatif relativement simple de gestion de sinistre traité par la société GMC. En effet, chaque domaine de risque peut avoir ses spécificités, notamment le temps écoulé entre la réception de la mission et la clôture du dossier. Celui-ci peut grandement varier de trois mois à trois ans pour les sinistres qui impliquent la responsabilité civile de tiers.

Un sinistre plus communément appelé un dossier, est l'équivalent d'une affaire pour des activités commerciales, il correspond à l'historisation des informations entre le gestionnaire et le mandataire.

Nous constatons de plus en plus que les sinistres dans leur globalité sont gérés comme un projet avec des ressources disponibles, un budget, des dates de fin et des recettes intermédiaires. Ainsi les besoins d'anticipation sont nombreux et reposent sur la puissance du logiciel qui doit y répondre.

Le processus chronologique débute dès la réception de la mission (1), où le responsable des affectations va en fonction du domaine et des spécialités des experts, affecter la gestion du sinistre (2).

L'assistante et l'expert en charge du dossier vont collaborer afin d'avancer simultanément à différents niveaux de leurs missions. L'assistante va typiquement établir l'accusé de réception de la mission (3) qui a pour vocation de signaler au mandataire que celle-ci est acceptée. Par la suite l'expert fait une première demande de pièces (4), et l'assistante va compléter la démarche en demandant des pièces plus précises (5) ou en faisant des relances (6).

Ensuite nous entrons dans un processus itératif où l'expert commence à traiter les pièces reçues (7) et en tire les conclusions qui seront présentes dans le rapport d'expertise (8). L'assistante communique aux différentes parties les informations sur l'avancée du dossier (9).

Des étapes clés dans la gestion de sinistre apparaissent et sont bien connues des différents pôles. Le plus souvent le but est d'arriver le plus rapidement possible au rendez-vous d'expertise (10), suivi par le rapport d'expertise qui est le rapport final sur le dossier (11). Après validation de celui-ci par le responsable des affectations (12), le dossier peut être expédié (13) et facturé (14).

Il peut arriver des situations de conflit (15), dans ce cas le dossier entre dans un statut particulier, où le responsable des experts va tenter de le résoudre. Ceci est le pire des scénarios, et existe seulement dans les dossiers les plus compliqués, où le conflit peut se situer, entre les parties, entre l'expert (plus globalement le cabinet d'expertise) et une des parties. Cela retarde voire annule la facturation possible de ce dossier.

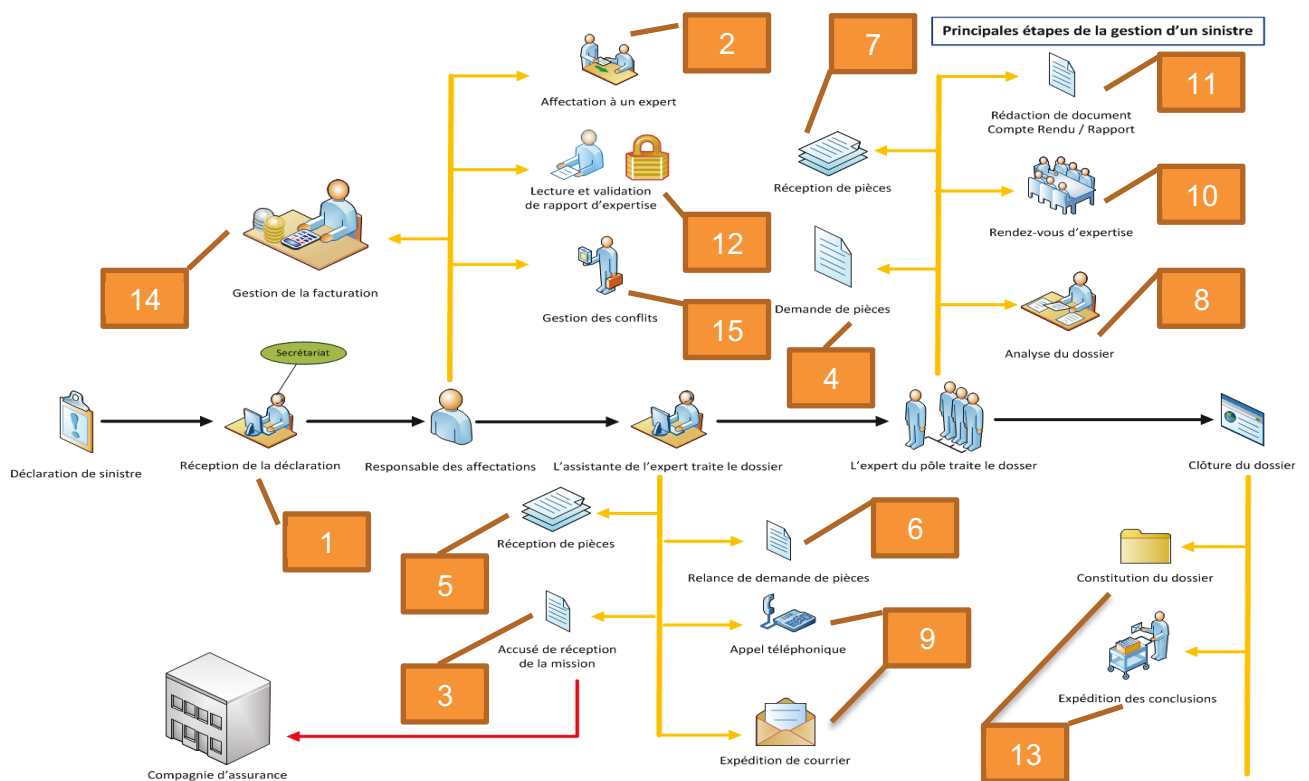


Figure 3 – Métier de Gestion de sinistre

Comme nous l'avons vu dans cette section, le métier du groupe se caractérise par un besoin de travail collaboratif important, avec des contraintes de qualité de services et de sécurité des données.

C'est pour cette raison que le système d'information a une place importante dans tous les processus, et sert notamment à automatiser les actions répétitives, à fournir des indicateurs d'alerte et à respecter les règles métier.

1.2 Le système d'information

Le groupe n'a jamais souhaité acquérir une solution logicielle présente sur le marché pour répondre à ses besoins pour plusieurs raisons :

- devoir de confidentialité et de maîtrise des données,
- coûts d'acquisition élevés pour couvrir l'ensemble des activités,
- réactivité d'évolution trop importante face aux contraintes,
- projet de créer sa propre société éditrice de logiciel.

La [Figure 4](#) montre de façon synthétique l'étendue des solutions qu'Insiide (ancien service informatique) maintient à l'heure actuelle. Celles-ci tendent à couvrir l'ensemble des activités en faisant preuve d'adaptation continue.

Le système d'information s'appuie sur des applications métiers spécifiques aux activités principales (GMCSOFT pour GMC, SESSOFT pour SES) et des applicatifs partagés qui tendent à couvrir les activités secondaires d'édition de logiciel déployés par les dirigeants depuis quelques années (GRCLIVE, GBM et OPENAVOCAT). La troisième partie du système est les outils décisionnels utilisés à des fins de gestion de l'activité principale GMC et SES.

- les deux principales applications du groupe : GMCSOFT et SESSOFT
- les activités secondaires : GRCLIVE, GBM, OPENAVOCAT
- les analyses et les statistiques : outils décisionnels.

Les deux principales applications

Les deux applicatifs GMCSOFT/SESSOFT sont essentiels au maintien de l'activité principale (GMC/SES), productrice du chiffre d'affaire annuel. C'est pour cette raison qu'ils doivent être performants, stables et robustes afin d'assurer le travail collaboratif des utilisateurs.

Les logiciels permettent de travailler sur trois fonctionnalités directrices :

- stocker l'ensemble des informations traitées par les collaborateurs dans un système unique,
- assurer un flux de travail collaboratif et standardiser la production de chacun,
- analyser les données pour aider les processus décisionnels.

Activité secondaire

L'activité secondaire comprend l'édition de deux autres applicatifs GBM et GRCLIVE en production à l'heure actuelle. Ils sont respectivement destinés à un cabinet d'expert-comptable et une filiale d'un fournisseur d'accès à internet historique spécialisé dans les solutions pour les professionnels. Ces deux clients sont générateurs de demandes d'interventions fréquentes qui constituent donc une charge de travail incompressible relativement prévisible. Nous devons donc en tenir compte lors de toute planification afin de respecter, d'une part nos contrats actifs, et d'autre part d'établir un calcul au plus juste.

OPENAVOCAT, logiciel collaboratif de gestion de dossier juridique n'a jamais dépassé le stade de prototype malgré le nombre d'années depuis lequel la direction souhaite avoir un produit de ce type à son catalogue. Son évolution n'est pas intégrée au projet global dont il est question mais il fait partie intégrante de l'étude de l'existant et des contraintes à prendre en compte. Par ailleurs il est fort probable que sa finalisation soit envisagée lorsque le projet en cours sera achevé.

Analyses et statistiques

La brique présentée en [Figure 4](#) est indispensable au suivi de l'activité et à la prise de décision : l'analyse et les statistiques. Nous avons plusieurs outils disponibles, en interne à la direction, ou en externe à des grands comptes et partenaires externes. Ceux-ci doivent faire preuves de fiabilité quand à ce qui est exposé car des choix, des paiements, des reconduites de contrats sont actés par les différentes analyses qui sont effectuées.

Cet outil nous procure une image de marque solide qui rassure les clients qui sont amenés à nous déléguer la gestion de flotte de contrat d'assurance. En effet, l'expérience montre qu'une gérance de masse est plus efficace quand des indicateurs nous permettent d'adapter nos actions à la situation.

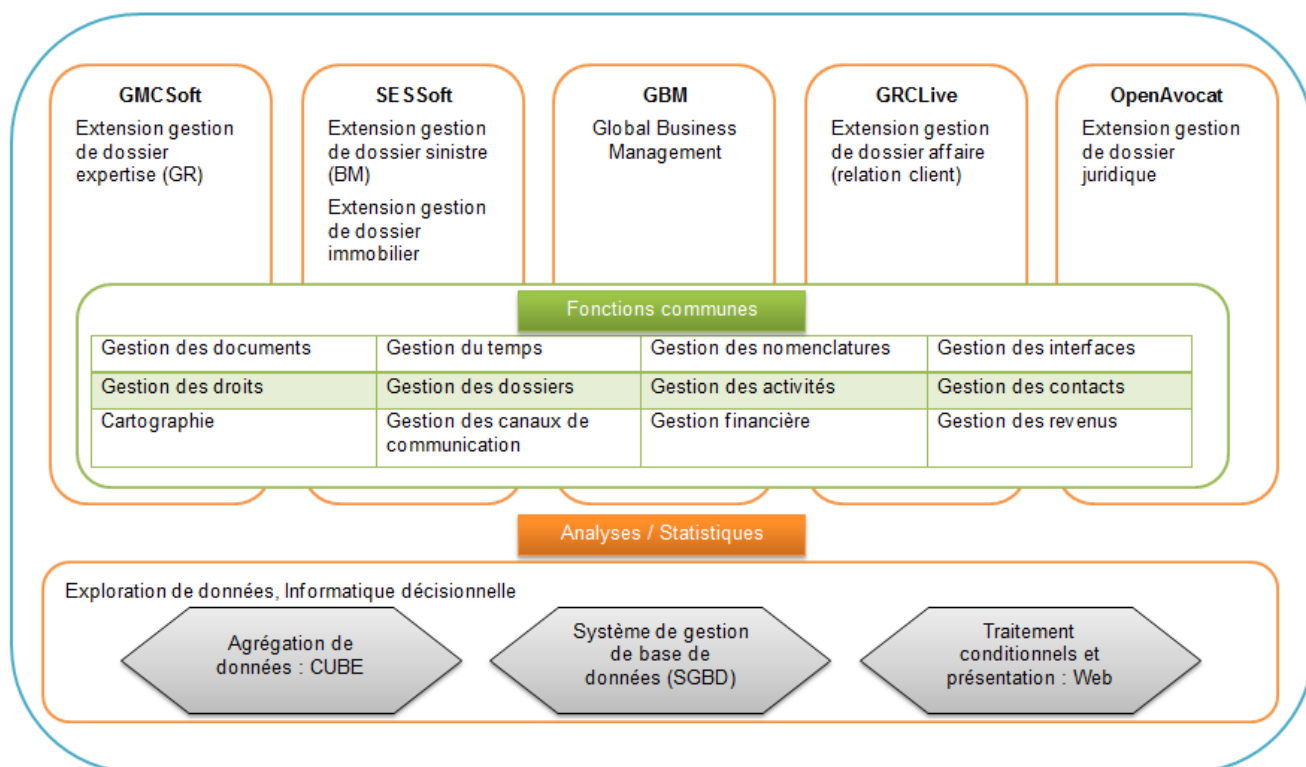


Figure 4 – Suite logicielle existante

1.3 Présentation générale du projet

Au sein du groupe, la [Figure 1](#) montre qu'Insiide est une structure à part entière qui intervient dans toutes les activités du groupe. Elle est, à ce titre, l'unique intervenant de la création du système d'information adapté aux besoins. Il existe cependant un paradoxe, elle est indispensable mais reste une entité ayant des prétentions externes, cependant elle est aussi complètement dépendante notamment d'un point de vue financier.

Les collaborateurs du service informatique, regroupés dans une structure spécialement créée, couvrent donc l'ensemble des demandes, avec la partie infrastructure système et réseau puis la partie logicielle. Cette entité, composée de cinq collaborateurs ([Figure 5](#)), est spécialisée dans les solutions de gestion d'expertise et de gestion de sinistre.

La suite logicielle¹ telle qu'elle est représentée [Figure 4](#) regroupe l'ensemble des logiciels conçus. Ces derniers répondent aux besoins des activités du groupe ainsi que ceux livrés à des clients externes.

L'évolution d'Insiide, [Figure 5](#), a débuté dès la mise en place du projet présenté dans ce mémoire. Plusieurs modifications ont été opérées en ce qui concerne les tâches et missions de l'ensemble de l'équipe informatique. Me concernant, mes affectations ont donc été entièrement modifiées, elles se sont en effet diversifiées en intégrant de nouvelles dimensions. Ceci sera détaillé plus précisément dans la suite de mon écrit.

Avant le projet

Recruté en 2009 par Insiide ([Figure 5](#)), j'ai dans un premier temps rejoint l'équipe de développeur sur la suite logicielle de l'ensemble du groupe ([Figure 4](#)). En cours d'année 2010, j'ai pris en charge pour environ trente pourcents de mon temps de travail, l'administration système et réseau ainsi que la gestion du parc informatique. J'ai travaillé dans cette optique avec l'administrateur réseau qui lui-même est chargé en parallèle de répondre aux besoins d'analyses et de statistiques ([Figure 4](#)). Nous avons mené un travail en collaboration complémentaire et productif afin de nous répartir les tâches selon les priorités de chacun.

¹ Une suite logicielle est un ensemble de programmes informatiques, offrant des fonctionnalités liées, et possédant souvent une interface graphique plus ou moins similaire.

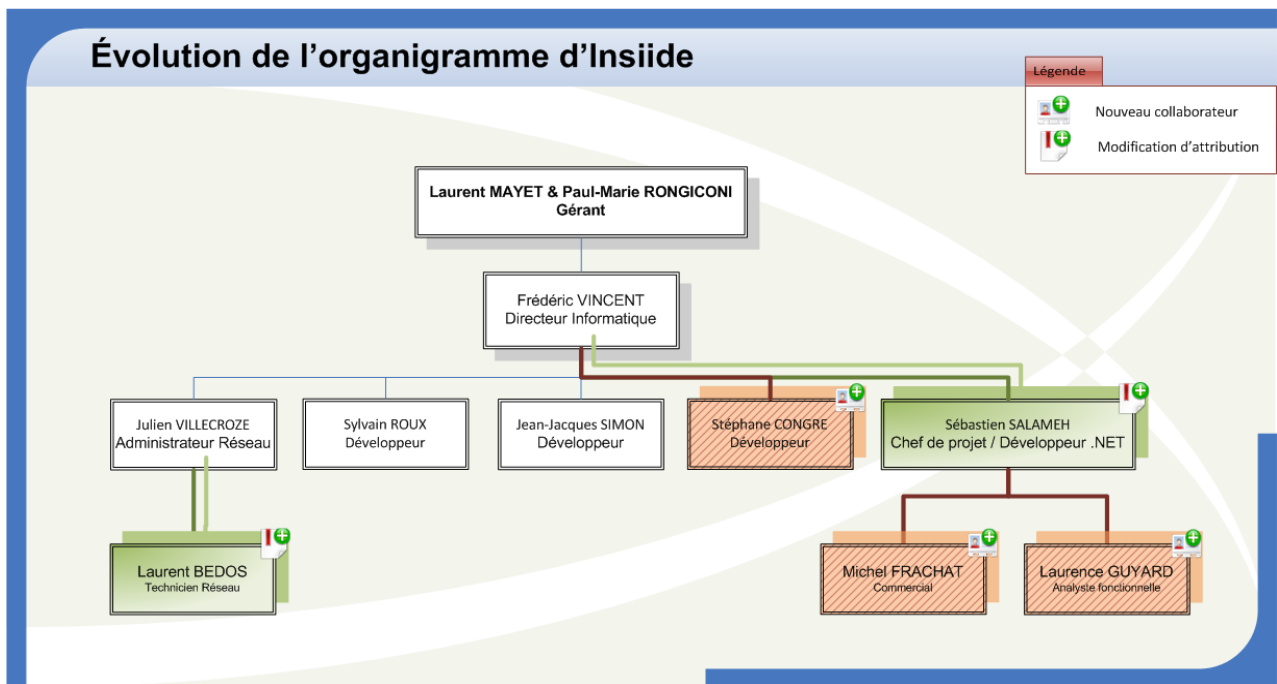


Figure 5 – Évolution de l'organigramme d'Insiide

Au démarrage du projet

Mes démarches dans la veille technologique et les nouveautés m'ont amené à me spécialiser dans les outils et méthodes destinés à améliorer le système d'information dans son ensemble. Ma polyvalence m'a progressivement permis d'entamer des actions plus ciblées sur la recherche et développement que sur la maintenance logicielle. C'est à cette occasion que de nouvelles technologies, notamment le dotNet² sous une forme simplifiée, ont fait leur entrée chez Insiide. Cela a débuté par l'amélioration des outils décisionnels à l'usage de la direction.

À partir de la mise en place du projet

En fin d'année 2010, c'est-à-dire un an après mon arrivée, un véritable changement d'attribution a été mené afin d'aboutir à la création d'une branche inexistante dans le service : Pôle de recherche des applications Microsoft dotNet, en parallèle du langage historique utilisé au sein du groupe, Delphi³.

L'intérêt de ce pôle est d'intégrer progressivement les nouvelles technologies avec une recherche ciblée sur leurs avantages et inconvénients appliqués à notre domaine d'activité. Ainsi nous pouvons assurer une maintenance préventive plutôt que curative une fois dépassé par les événements.

Mes attributions et difficultés découlent du cadre général fixé au préalable. Ce dernier est composé de plusieurs axes pour mieux évaluer la source et les implications :

- responsable du cadre d'action technique pendant le déroulement du projet,
- référent de la communication avec les clients d'Insiide, interne et externe.

Mes implications sont compatibles avec la polyvalence mise en œuvre dans l'entreprise. Tout d'abord il y a la possibilité d'intervenir sur différents aspects techniques tel que les applicatifs internes, mais aussi l'environnement système, serveur, réseau, et le parc informatique dans son ensemble. Ce dernier m'apporte une vue, en tant que développeur logiciel, différente et m'incite à conjuguer les évolutions de développement et environnementale dans le même temps. Il s'agit de prendre en compte un certain nombre de contraintes lors d'une modification afin d'en dégager les implications précises.

La décision de m'affecter à ce projet est également issue du fait de mon engagement dans une série d'évolution qui m'a mis en contact direct avec le client (utilisateur interne ou externe). Tout ceci a des fins de définition du besoin, cadrage, et apport de solution. Ce genre d'exercice à échelle beaucoup plus modéré constitue plusieurs expériences dans les contextes professionnels rencontrés.

² Microsoft .NET est un ensemble de produits et de technologies édité par Microsoft. Il permet notamment de développer en plusieurs langages de programmation.

³ Delphi est à la fois un environnement de développement intégré et un langage de programmation orienté objet. Édité par Borland il implémente la version orientée objet du Pascal.

En effet chaque entité va cumuler un historique et un milieu qui lui est propre. La maîtrise de cet environnement représente une plus-value dans une gestion de projet d'envergure comme celui-ci.

C'est dans ces fonctions que j'ai mis en œuvre en autonomie de conduite et de réalisation du projet :

- l'organisation des transferts de compétences de mes précédentes attributions,
- l'organisation efficace du service (l'équipe de trois développeurs d'Insiide),
- l'organisation de la communication entre Insiide et le reste du groupe,
- l'organisation de la communication entre Insiide et ses clients.

Le projet présente donc mes deux contributions :

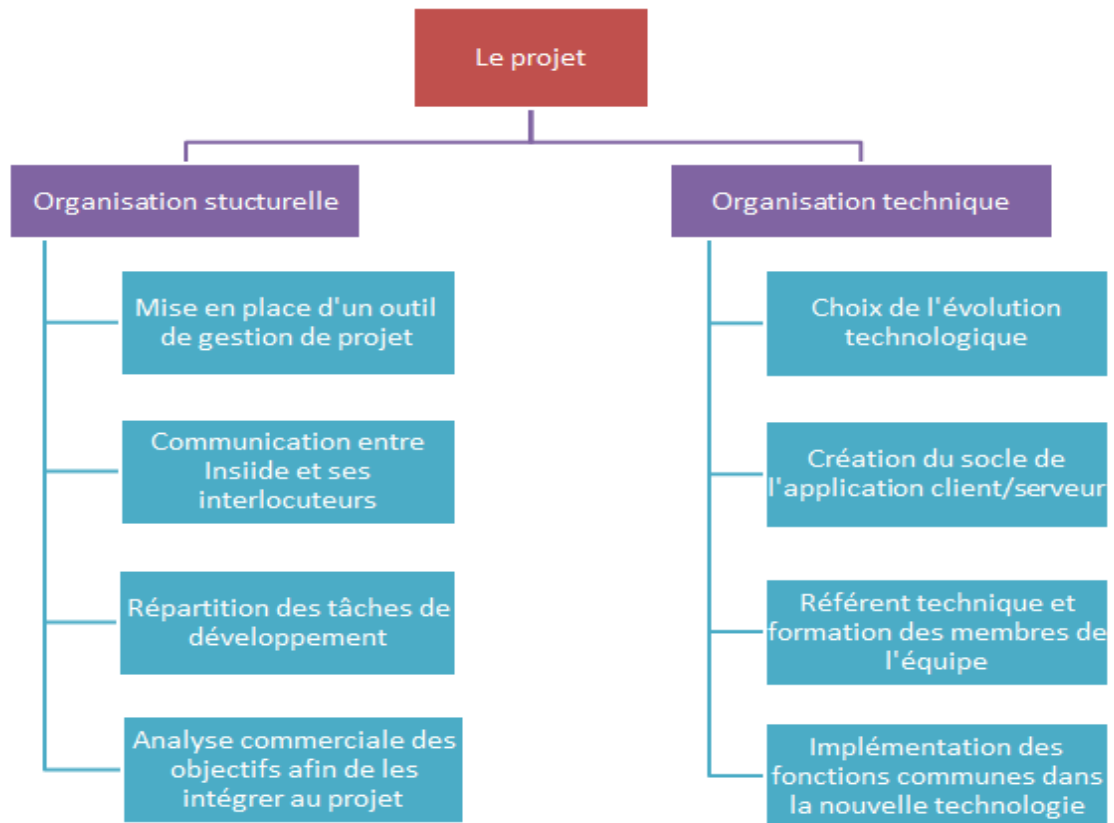


Figure 6 - Les deux composantes du projet

1.4 Ligne directrice du projet

Insiide est à une étape charnière de son existence. La multiplication des activités métier, la nécessité de mutualiser et d'obtenir un retour sur investissement sont à l'origine de la naissance du projet.

Les limitations budgétaires, techniques, liées aux compétences et au savoir-faire de chacun et le manque d'innovation sont également la cause d'une inertie voire même d'une incapacité à répondre à certains besoins. Insiide doit également rentabiliser son activité financée jusqu'à maintenant par Group Mayet Consultant. Il sera nécessaire d'attendre les premiers retours financiers de ce projet, ainsi, cette notion de rentabilisation fait partie des objectifs directs.

Face à cette ligne directrice, les solutions consistent à utiliser plus efficacement les compétences de chacun. Du point de vue du système d'information, l'interopérabilité totale entre les technologies est la clé de la réussite.

Le but consiste en la création d'un système unique qui gère tous les besoins du groupe au sens le plus large du terme, c'est-à-dire l'ensemble des activités de production.

Nous nous sommes posé les questions suivantes pour répondre aux attentes sans omettre les contraintes liées à chacune d'elles.

- Comment mettre à disposition un service à un client en quelques manipulations ?
- Comment utiliser le même socle pour des applications différentes, client léger⁴ et client lourd⁵ et dans des langages différents ?
- Comment assembler dans un logiciel le savoir-faire acquis depuis des années ?
- Comment réutiliser les développements existants comme base d'intégration des fonctions communes ?

La tendance avant la mise en place du projet nous montre que le système d'information n'évolue pas. Bien au contraire, tous les efforts servent à maintenir la qualité de service à un niveau acceptable.

L'objectif est donc d'inverser cette tendance de deux façons :

- introduire la notion de recherche et développement et d'innovation,
- faire évoluer les produits actuels et en créer de nouveaux.

Group Mayet Consultant et Insiide ont donc respectivement tracé leur avenir afin d'intégrer à leur échelle les pressions budgétaires qui pèsent sur les sociétés. Tout ceci, a pour but de faire croître le groupe sur des secteurs où la concurrence est déjà forte et où la nécessité de faire preuve d'innovation est de plus en plus importante.

⁴ Il existe plusieurs sortes de client léger mais ici nous parlons de client léger pour un logiciel qui s'exécute dans un navigateur web.

⁵ Un client lourd est un logiciel qui comprend l'ensemble des fonctionnalités et qui n'a besoin d'un serveur que pour stocker les données. Il embarque donc les interfaces graphiques et l'intelligence applicative.

1.4.1 Du point de vue du groupe

Group Mayet Consultant est bien le principal moteur de ce projet car ces objectifs sont :

- développer l'entité éditrice de logiciel pour qu'elle soit autonome puis génératrice de bénéfices qui amortiront l'investissement effectué depuis des années,
- unifier les développements sous des domaines fonctionnels,
- assurer la pérennité de la suite logicielle utilisée par le groupe.

Développement de l'entité éditrice de logiciel

Un des principaux objectifs est le développement de l'entité éditrice de logiciels, Insiide. Ceci afin de positionner sur le marché des produits et services éprouvés et améliorés en interne depuis des années. Le but n'est pas de commercialiser GMCSOFT ou SESSOFT car la direction ne souhaite pas doter ses concurrents des logiciels de gestion de sinistre. Cependant le savoir-faire acquis en gestion interne d'une entreprise qui peut se recouper sur des termes génériques peut être conditionné pour être distribué sur d'autres secteurs que l'on jugera accessible.

Unification des développements

L'unification des développements a permis de dégager plusieurs domaines fonctionnels qui sont couverts par les logiciels développés en interne. Certains modules sont extrêmement poussés et assurent une interaction totale avec le système d'information. D'autres ne le sont pas spécialement, soit parce que le besoin ne s'est pas fait sentir, soit parce qu'il a été jugé de priorité basse.

Ainsi une tâche complexe est la faculté d'expression de chacune des propriétés et de leur état d'avancement, afin d'avoir une idée précise de l'existant ([Annexe I Couverture fonctionnelle page 57](#)). Cela est directement en ligne pour la création de nouveaux produits ou fonctionnalités aux actuels utilisateurs et futurs clients.

Pérenniser la suite logicielle existante

La pérennisation de la suite logicielle consiste à accroître la durée de vie de la suite logicielle afin qu'elle puisse toujours répondre au besoin de travail collaboratif du groupe.

L'enjeu transversal est la réalisation de certaines attentes de plus en plus nombreuses et urgentes, occultés avec les technologies actuelles. Parfois certaines remettent en cause tellement de paramètres qu'un changement complet de technologie ne s'avère pas plus compliqué qu'une refonte d'un système vieillissant.

Ce débat est également animé par une réflexion rétrospective et la pratique de l'informatique de manière générale. Le pôle d'expertise informatique de GMC a l'opportunité de gérer de nombreux sinistres dans le domaine technologique et scientifique. Cette appréciation est très présente dans les décisions prises et apporte un angle de vue différent.

1.4.2 Du point de vue d'Insiide

Insiide est, pour l'instant, encore sous tutelle du groupe, cependant elle a des objectifs qui serviront ceux du groupe mais aussi son extension :

- réduction des coûts de développement et des opérations de maintenance,
- investissement dans la recherche et développement,
- qualité de services.

Réduction des coûts de développement

L'objectif premier est la réduction des coûts de développement, plus précisément la diminution des temps de développement. En effet, ce dernier est le paramètre qui va le plus rapidement faire percevoir les effets d'une simplification du processus de déploiement de fonctionnalité. Mais pas seulement, derrière cette rationalisation il y a également la facilité d'administration d'un système complexe, ouvert, et visant de multiples horizons.

La restructuration technologique assurera l'interaction avec l'ensemble de nos applications. Qu'elles soient client léger ou client riche le but est d'obtenir un lien permanent et unique entre les applicatifs clients et l'applicatif serveur. La prise en charge de l'environnement système et réseau n'est pas non plus à négliger, les impacts sont à appréhender pour une administration standardisée.

Réduction des opérations de maintenance

La maintenance est au centre du projet, le but est de réunir en un point le cœur des programmes. Les serveurs sont à dimensionner en rapport de leur sollicitation et de ce qui est hébergé sur chacun d'eux. Toutes les sécurités devront être prises pour conserver une continuité de service. Nous allons donc avoir une approche modulaire et plus robuste (client/serveur⁶).

Investissement dans la R&D

La recherche et développement est non négligeable, elle concerne même jusqu'à trente pourcent du projet. Celle-ci est donc à prendre en compte dès la planification et le budget prévisionnel. Cet investissement couvre deux axes, dans un premier temps il assure le choix d'une nouvelle technologie dans le système d'information. Puis dans un second temps il permet d'anticiper les problèmes techniques et les corrections à mettre en œuvre pour optimiser nos réalisations.

L'objectif est de parvenir à une suite logicielle qui répond le mieux possible à tous les besoins actuels. La multiplication des plateformes d'exécution (fixe et mobile) et les différents modes de communications qui se développent sont des nouveaux besoins en termes d'utilisabilité. Nous devons anticiper l'évolution suivie pour la suite logicielle et plus largement le système d'information.

S'inscrire dans un niveau de qualité de service

Cette évolution technologique, aussi bénéfique qu'elle puisse être, ne doit pas faire subir de coupure de service sur les plateformes de production. Les clients et les utilisateurs finaux ne sont pas directement concernés par ces améliorations tant qu'elles ne leur apportent pas de nouvelles fonctionnalités. L'avant-projet et les restructurations techniques ne sont donc pas une plus-value fonctionnellement parlant. Le public visé dans ce projet ne correspond pas directement aux usagers actuels des services, se sont surtout le potentiel de nouveaux clients que va représenter l'ouverture d'opportunité. À terme, ce qui est intéressant pour les utilisateurs actuels, ce sont les fonctionnalités évolutives qui vont apporter une collaboration encore plus forte.

Bien entendu, les apports non fonctionnels ont un objectif sérieux, celui d'une maintenance applicative simplifiée et par ce fait une charge de travail en baisse. Insiide va donc avoir la possibilité d'assurer un niveau de qualité de service pour l'ensemble des solutions qui seront à son catalogue.

⁶ Une approche client/serveur consiste à scinder les rôles client et serveur en deux applications distinctes. L'une (le client) envoie des requêtes, l'autre (le serveur) attend les requêtes des clients et y répond.

2. Réorganisation structurelle

La réorganisation structurelle d'Insiide mais aussi de Group Mayet Consultant est un prérequis à la réorganisation technique du système d'information. Même si celle-ci n'est pas, en terme de temps passé, aussi importante que l'évolution technologique, elle a justement permis à cette dernière d'être réalisée. De plus il me semble important de préciser les différentes restructurations effectuées car elles sont de toute façon, même sans concrétisation technique, une plus-value pour la gestion du système d'information.

Chronologiquement nous avons agi sur l'organisation d'Insiide avant de procéder à celle du groupe. En effet, la plupart des aménagements visent à améliorer le processus de réalisation et doivent être réalisés avant toute extrapolation.

2.1 Insiide

Insiide a besoin d'être réorganisé car comme nous pouvons le voir sur la [Figure 7](#) nous avons plusieurs problèmes. Nous pouvons mettre en évidence les demandes qui arrivent toutes avec la même importance, priorité, criticité, à Insiide ce qui représente la principale source des lacunes. Par la suite, et dû à cette méthode de travail, chaque membre de l'équipe va prendre des décisions, en concertation avec d'autres personnes ou pas, pour apporter une réponse à une question. La perception de chacun pouvant varier, le manque de communication évident et la quantité de demandes sont autant d'éléments qui ont pour conséquence le manque de fiabilité des développements.

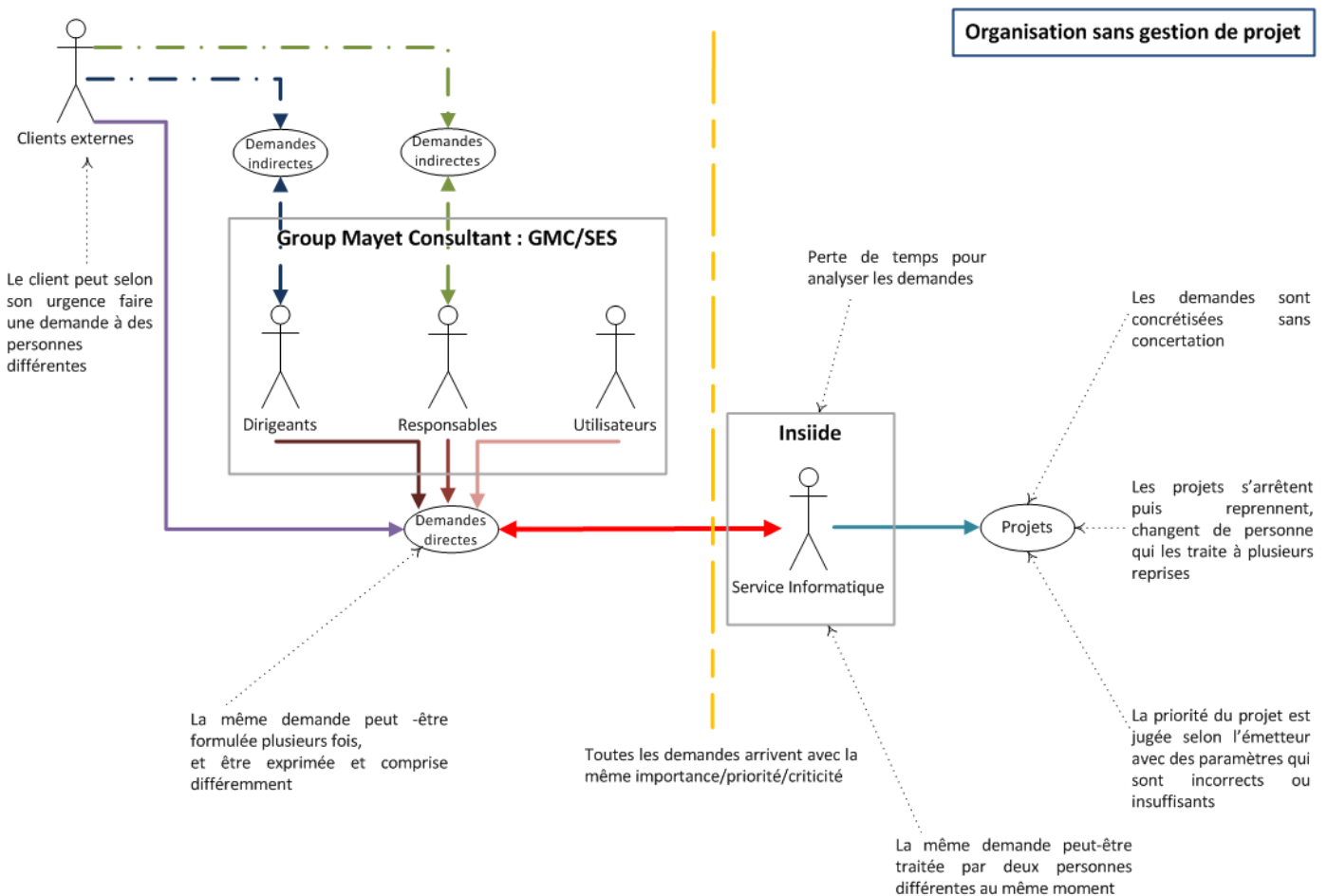


Figure 7 – Organisation d'Insiide sans gestion de projet

Avec l'apparition et la mise en place de ce projet nous avons répartis nos efforts et notre temps de travail, non seulement sur les activités historiques (principales et secondaires), mais aussi sur les nouvelles évolutions nécessaires.

Ces deux objectifs sont donc concrétisés avec différentes phases mises en œuvre qui consistent à créer de nouveaux produits tout en consolidant la suite logicielle existante.

L'outil de gestion de projet

Mis en place, il est la base de toute l'organisation. Sans lui nous serions incapables de déceler dans un premier temps là où nous devons agir pour réduire la charge de travail afin d'assurer le développement de produits finis et commercialisables. Dans un second temps la gestion de projet a également pour but la clarification des différentes priorités car nous nous sommes aperçus que la perception des priorités était trop aléatoire selon les développeurs.

Par l'intermédiaire de celui-ci l'ensemble des demandes y sont archivées et les ressources humaines enregistrent leurs temps passés sur chacune d'elles. Cela met en avant le travail d'équipe sur une demande qui peut être à la base formulée simplement mais qui implique en réalité un travail plus important.

La recherche et développement (R&D)

Elle tient une place importante dès le début du projet, puis tend à conserver un maximum de régularité afin d'entamer une démarche continue d'innovation et d'amélioration de nos développements.

La R&D tend même à augmenter en fin de projet dans l'objectif de préparer l'après projet et la suite logique des développements.

Le développement de nouveaux produits

Il est directement issu des étapes précédentes afin de pouvoir planifier leur réalisation et faire des choix techniques au plus juste. Le commercial chargé de concevoir un catalogue produit a en effet besoin d'anticiper les différentes étapes et de savoir à quel moment le produit sera disponible à la distribution.

À la fin du projet nous allons avoir un nouveau logiciel commercialisable qui sera éventuellement décliné en plusieurs spécialités pour augmenter le potentiel de cible. Nous avons donc besoin d'exprimer en termes de fonctionnalité ce qui est couvert par nos logiciels afin que les commerciaux soient capables de comprendre les activités que nous pouvons couvrir.

Afin d'éviter des situations d'incompréhension mutuelle, entre les aspects commerciaux et techniques, j'ai tenu à expliciter de façon intelligible les directions que nous prenons et du résultat final attendu.

L'évolution technologique de la suite logicielle existante

Elle doit à terme permettre de diminuer la charge de travail issue des demandes effectuées par les sociétés internes (GMC/SES) au groupe. Effectivement la principale charge de travail d'Insiide reste la maintenance de la suite logicielle existante même pendant la réalisation de ce projet avec deux ou trois développeur à temps plein dessus selon les périodes. Ce qui montre bien que les premières évolutions effectuées doivent être bien réfléchies afin qu'elles représentent une réelle diminution de la charge de travail pour Insiide. Ceci afin d'allouer progressivement de plus en plus de ressources à la réalisation du projet.

La maintenance de ces deux logiciels GRCLive et GBM, représente une charge de travail minimale et constante sur la durée du projet. Ils sont inclus dans l'évolution de la suite logicielle existante mais ne sont pas prioritaire sur les activités principales du groupe GMC/SES. Concrètement si nous devons faire un choix, à implication de ressources égales, nous choisirons celui qui d'un point de vue financier est le plus rentable et/ou les retours sont les plus importants pour le groupe. En effet, en termes de source revenu, les activités secondaires ne représentent pas un chiffre d'affaire suffisamment important même pour assurer le fond de roulement d'Insiide. Cependant d'un point de vue commercial ce sont les deux seuls clients de cette dernière et nous faisons le maximum pour les conserver car l'idée et de développer notre activité auprès de ces entreprises en apportant régulièrement de nouveaux services.

La réorganisation structurelle a été appliquée à Insiide dans un premier temps, mais le groupe dans son ensemble est également concerné. Il est forcément intégré à ce projet car Insiide va comme nous l'avons vu réduire son investissement en ressource et en temps sur la maintenance applicative, tout en apportant des réponses de fond aux évolutions nécessaires.

2.2 *Group Mayet Consultant*

Le groupe composé des structures GMC, SES, SES Immo possède chacune un directeur et des responsables, le tout encadré par la direction générale ([Figure 1](#)).

À l'échelle du groupe

La direction est consciente de la réorganisation qu'il faut mener à l'échelle du groupe. Nous avons proposé lors d'une réunion préparatoire, la nomination par la direction, des responsables métiers pour chacune des activités. Ma tâche consiste à déterminer les personnes qui seraient les plus efficaces dans ces fonctions tout en maintenant leurs métiers d'origine.

Nous allons également faire appel à des professionnels externes, car le développement de nouveaux produits implique de les vendre. Insiide, n'a pas cette compétence, et a à plusieurs reprises eu des expériences négatives avec des commerciaux recrutés. Le but par le biais de ce partenariat et d'avoir une force de vente sur un secteur précis déjà connus par des spécialistes.

Des responsables métier ont été nommés

Ils sont nommés pour chacune des structures internes mais aussi pour les activités secondaires ([Figure 1](#)). Leur rôle est de rassembler les demandes de chacun des utilisateurs de son corps de métier, d'en vérifier la validité et la cohérence par rapport aux objectifs généraux et dans ce cas les remonter au chef de projet informatique. Ainsi nous allons considérablement réduire le nombre d'interlocuteurs potentiel une fois cette méthode appliquée à contrario du mode de fonctionnement actuel.

Nous pouvons également préciser que des réunions avec les responsables métier seront organisées. De périodicité hebdomadaire, les échanges sur les demandes à l'ordre du jour, vont permettre de définir les priorités et les échéances de chacune d'elles. Leur dernière attribution est d'effectuer la recette fonctionnelle d'un développement avant la propagation à l'ensemble des utilisateurs. Nous entrons dans un mode de projet essentiel, garant de la qualité requise pour les applicatifs métiers.

Des ressources supplémentaires

Elles entrent en interaction avec l'équipe d'Insiide de deux façons :

- la première est le nouveau collaborateur qui vient renforcer les développeurs, sur le développement d'application de la suite actuelle et à venir ([Figure 5](#)),
- la seconde est l'intégration de deux professionnels sous ma responsabilité qui sont présents pour commercialiser le nouveau produit que nous allons développer. Cela correspond à l'accord passé avec la direction. Ces derniers connaissent un secteur d'activité et ont des contacts concrets pour le nouveau produit à réaliser. Insiide peut développer celui-ci et souhaite devenir un acteur dans l'édition de logiciel. Nous aurons donc une collaboration étroite pour sa réalisation. Cependant ce nouveau produit est financièrement insuffisant pour rentabiliser Insiide, c'est la raison pour laquelle GMC/SES reste la priorité.

Le support utilisateur

De manière générale les utilisateurs devront également modifier leur rapport avec les collaborateurs du service informatique car toutes les demandes devront être filtrées en amont par les responsables métiers. Cependant certaines fonctionnalités vont résoudre des problèmes de fonds avérés et de ce fait limiter les demandes utilisateurs. Le but n'est pas de couper purement et sans réflexion le support utilisateur qui peut avoir des répercussions immédiates sur la production du groupe. Nous avons la tâche d'harmoniser les demandes avec une vision métier globale et cohérente sur le plan de l'informatique.

Après les différentes restructurations organisationnelles appliquées, le mode de communication à complètement été revu. Il gravite désormais autour d'un comité de pilotage où siège bien entendu la direction du groupe (dirigeants de GMC/SES), la direction informatique (le Directeur Informatique d'Insiide), les responsables métier (responsable de chacune des activités) et le chef de projet (moi-même). Chacun des acteurs à un rôle bien défini ce qui assure une cohérence dans la réalisation de projet informatique en prenant en compte l'avis de tous les responsables de branche. La direction de Group Mayet Consultant siège parfois aux réunions hebdomadaires pour s'assurer ou rappeler, si nécessaire, les priorités financières par rapport aux décisions qui peuvent être prises.

Il est également important de préciser que nous avons fait le choix d'une restructuration interne sans perturber ou modifier le fonctionnement et les modes de communications avec nos clients existants. Même si j'ai canalisé un certains nombres d'interlocuteurs, certains font toujours appel à leurs connaissances, tout en m'assurant que ces derniers soient représentés au comité.

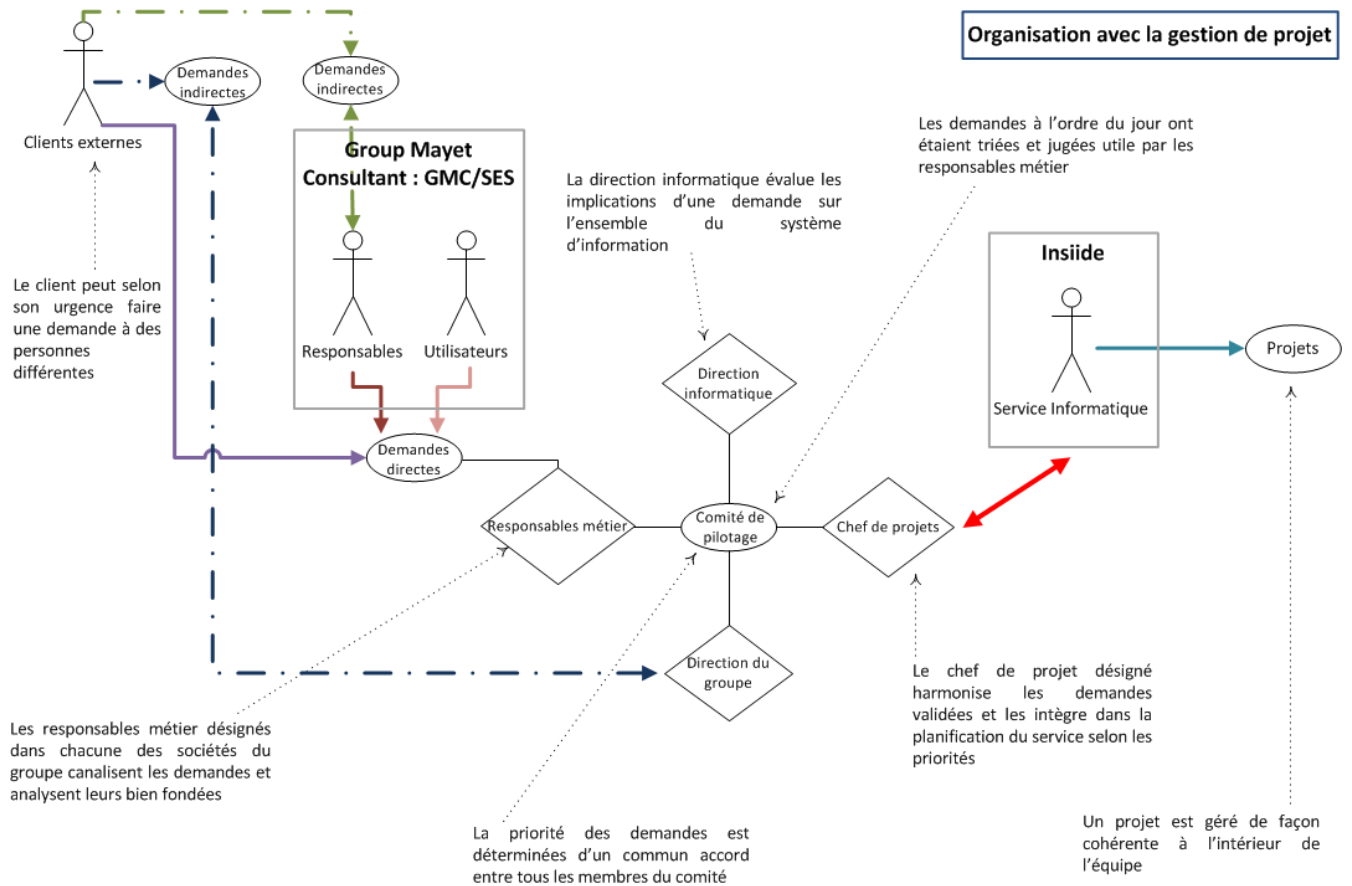


Figure 8 – Organisation d'Insiide avec gestion de projet

Sur le fond ces restructurations devront permettre de nous axer principalement sur le développement de nouveaux produits. Nous devons réussir à diminuer nos temps passés pour GMC/SES car les sociétés du groupe ne seront jamais des clients à part entière. Nous aurons toujours une relation différente à l'intérieur du groupe qui ne rapportera pas réellement de chiffre d'affaire à Insiide.

Pour mener à bien cette réorganisation structurelle il a été essentiel d'intégrer différentes dimensions. Je fais allusion, à l'aspect managérial et commercial jusqu'ici absent des différentes démarches entreprises par Insiide. Celles-ci m'ont été affectées lors de mon changement d'attribution (présenté en [Figure 5](#)) afin de répondre aux nouveaux besoins. En effet, j'étais stratégiquement bien placé pour y répondre notamment grâce à ma polyvalence technique et ma vision étendue des métiers du groupe. J'ai eu l'occasion de participer et d'être en contact avec ceux-ci depuis le début de mon intégration dans l'entreprise.

La dimension managériale

Cette dimension est la principale nouveauté de l'évolution de mes fonctions au sein d'Insiide.

La conduite de projet s'étend des choix techniques à la répartition des tâches à l'intérieur de l'équipe. Durant les étapes de programmation des parties sensible j'ai été le principal acteur. Même si mon implication dans ce type de tâche tend à disparaître, je lui ai témoigné de l'importance afin de conserver une vue précise des contraintes.

L'organisation consiste comme nous l'avons vu à réorganiser les flux de communications, et je suis désormais à ce titre l'interface entre les demandes émises par les responsables métiers et l'équipe de développeurs. La planification et la nouvelle organisation et la gestion de projet sont les garants d'une mise en place progressive et efficace.

La dimension commerciale

La dimension commerciale fait également partie des nouvelles tâches qui me sont assignées. Par celles-ci j'entends le recueil du besoin de mes interlocuteurs. Comme je l'ai évoqué ce sont les responsables métiers du groupe mais aussi le nouveau commercial pour le développement du nouveau produit.

À cela s'ajoute la connaissance du métier, pour lequel nous éditons des logiciels, qui doit être suffisante et généralement importante afin d'apporter la solution qui convient.

J'ai aussi tenté de modifier la rigueur de développement afin d'avoir la possibilité de garantir une qualité de service. Dès maintenant, le but est d'accorder autant d'importance à nos clients internes qu'externes afin de pouvoir respecter nos engagements.

Enfin, mon rôle consiste également à étudier la notion de rentabilité par rapport à la réalisation d'une demande. Cela passe par une compréhension du besoin, une estimation des ressources nécessaires (principalement en temps de travail) afin d'en déduire le coût global de l'opération. S'il apparaît que celui-ci est trop important par rapport au bénéfice il sera exclu, repoussé ou analysé différemment. Je l'ai notamment rencontré lors de la réalisation du nouveau produit ou certaines fonctionnalités superflu, ne doivent pas être prioritaire car elles représentaient un effort trop important de l'équipe sans que celles-ci n'apporte au produit final et donc au client une plus-value significative. C'est souvent la vision du besoin conjugué avec la vision informatique qui permet de dégager les priorités.

C'est également pour cette raison que je ne peux pas lors de la réalisation participer de façon conséquente au développement, car il faut que je puisse garder suffisamment de recul pour faire la part des décisions à prendre. Le but est de trouver le juste milieu pour une demande entre les ressources impliqués et le bénéfice pour le client final.

De l'ensemble de ces constatations, nous allons planifier en premier lieu la refonte des parties déterminées comme source de contrainte avant toute autre étape. Nous nous attèlerons à ce chantier qui touche l'ensemble des technologies et qui représente le plus lourd investissement de ce projet.

3. Réorganisation technique

Les objectifs du groupe sont ambitieux à l'égard de ce projet. En effet, le contexte est assez complexe et n'est pas favorable aux évolutions techniques nécessaires. Nous faisons face à un système qui a été conçu il y a plusieurs années pour des besoins, objectifs et contraintes complètement différents de la situation actuelle. Celui-ci a tout de même évolué pour faire face aux exigences et priorités les plus hautes. Cependant ce mode de fonctionnement et la volumétrie de travail n'ont pas assurés la réalisation de projets de fond qui peuvent permettre à Insiide d'être auto suffisante.

Cela s'explique notamment par l'essor du groupe qui est en constante évolution chaque année. Les contraintes orientent les prises de décisions mais le groupe n'imagine pas le fonctionnement de son activité sans l'informatique ou avec un système qui n'évolue pas.

Les nouvelles demandes sur les logiciels finaux sont fréquentes et abondantes. C'est pour cela qu'il est nécessaire de scinder en différents points de vue les objectifs et contraintes fonctionnelles et techniques afin de justement mieux mesurer leur aspect contradictoire.

3.1 Fonctionnelle

Les objectifs et contraintes fonctionnels sont issus, d'une part de la réorganisation structurelle et d'autre part des objectifs stratégiques de Group Mayet Consultant et d'Insiide.

3.1.1 La suite logicielle existante

Concernant les produits internes qui vont évoluer à l'égard de ce projet, cela ne doit représenter aucune perturbation du point de vue fonctionnel pour l'utilisateur final. Aucune différence de comportement ne doit être décelée par ce dernier.

Généralités

La suite logicielle existante se verra implémenter progressivement les fonctionnalités déterminés comme prioritaires par les responsables métiers :

- des fonctions existantes qu'il est nécessaire de faire évoluer,
- de nouvelles fonctions qui renforceront le travail collaboratif.

L'aspect progressif du projet est extrêmement important, car cela implique l'interopérabilité de la nouvelle technologie avec la suite logicielle existante.

Ainsi en tant que chef de projet j'ai effectué des mises au point avec chaque responsable métier afin :

- d'intégrer leurs objectifs, leurs contraintes et leurs priorités dans la ligne directrice du projet,
- comprendre la raison pour laquelle une fonctionnalité est plus attendue qu'une autre.

Les nombreuses réunions avec les responsables métier, mais également dans certains cas avec des utilisateurs internes spécialisés dans un domaine d'activité précis, m'ont apporté du recul sur la gestion de projet.

La difficulté de ce projet se situe effectivement dans la faculté à apporter une solution qui conviendra à un maximum d'utilisateurs et de clients.

Nos clients actuels

Comme nous avons pu le voir sur la [Figure 1](#) le groupe à une activité secondaire qui peut être apparentée à des clients d'Insiide, même si à l'origine Insiide n'existait pas encore lorsque ces solutions ont été commercialisées. Un axe de progression consiste à augmenter le potentiel d'intervention auprès de ces clients.

Nous allons donc leur proposer différents modules qui peuvent s'intégrer à la solution logicielle existante. Je me suis donc rapproché de nos clients afin de faire le point de manière générale sur la solution qui est utilisée. Le but est de déceler, d'une part les améliorations que nous pouvons porter au logiciel GBM et GRCLive ([Figure 4](#)), et d'autre part les modules que nous pouvons proposer dès maintenant ou dans un futur proche afin de compléter les services qu'ils utilisent.

3.1.2 La création de nouveaux produits

Chaque nouveau produit développé pour de nouveaux clients fera l'objet d'un cahier des charges fonctionnel décrivant le fonctionnement attendu. Pour les métiers qui sont nouveaux pour Insiide, comme le produit associatif que nous allons développer, Insiide ne prendra pas en charge la réalisation du cahier des charges. D'une part nous n'avons pas les ressources suffisantes pour gérer autant de sujet en parallèle, d'autre part l'objectif n'est pas de nous diversifier sur les aspects métier au point de perdre notre savoir-faire historique.

Pour nous, un nouveau produit pourra être un progiciel, un module, une fonctionnalité qui sera développé et mise à disposition à un client. Le but est de multiplier les clients car nous avons déjà beaucoup de services à proposer une fois qu'ils seront unifiés dans une même technologie cohérente.

3.2 L'évolution de l'architecture

À terme, l'ensemble des fonctions communes, [Figure 4](#) doivent être portées dans la nouvelle technologie choisie. Elles devront être gérées d'une façon cohérente et organisée afin de permettre l'interopérabilité des technologies présentées sur la [Figure 9](#).

3.2.1.1 Illustration des contraintes technologiques

Généralités

L'architecture technologique illustrée sur la [Figure 9](#) nous montre comment sont organisées les technologies par rapport à la suite logicielle présentée en [Figure 4](#). Le système conçu est vieillissant et extrêmement complexe à manipuler.

Nous pouvons dégager des trois technologies les éléments qui se distinguent et l'objectif de chacun d'eux.

Les applications Windows appelées « client lourd » développées en Delphi sont les plus unifiées en termes de code source. Il y a une partie des fonctions et interfaces spécifiques à l'activité mais le tronc commun est identique.

Les applications Web, appelées « client léger » développées en PHP sont hétérogènes car les besoins sont entièrement différents d'une activité à une autre. Concrètement il y a eu duplication de code source ce qui entraîne une maintenance de plus en plus longue avec la multiplication de services durant des années. Le point commun de l'ensemble de ces « intranet » ou « extranet » réside dans l'exploitation de bases de données de production alimentées à quatre-vingt-dix pourcents par les logiciels de gestion « client lourd ».

Les applications Web, destinées aux statistiques sont un mélange de technologie selon le besoin qui consiste à manipuler les données (à l'aide de système ETL⁷) et les présenter à l'aide de client léger principalement. L'architecture doit être entièrement revue afin d'harmoniser cela dans une démarche commune de mutualisation de fonction, ainsi que pour avoir une technologie plus adaptée à l'édition de système décisionnel.

⁷ Extract-Transform-Load (ETL), extracto-chargeur, est un système permettant d'effectuer des synchronisations massives d'une source vers une autre, en général une base de données.

Forte interdépendances des modules

Les effets de bords sont nombreux, non connus, et inévitables lorsqu'on souhaite apporter une modification de fond. Ceux-ci se déclinent en deux parties :

- l'absence de synchronisation entre les différentes technologies utilisées,
- l'absence de répartition des activités du groupe du point de vue informatique.

Cela pose des problèmes de compatibilité et renforce les difficultés de gestion de projet afin de parvenir aux évolutions sur la suite logicielle sans perturber le fonctionnement actuel.

L'absence de synchronisation entre les différentes technologies utilisées

Nous avons rendue interopérable trois socles technologique, [Figure 9](#) :

- la partie logicielle développée sous Delphi,
- la partie web développée principalement en PHP⁸,
- la partie « statistiques » utilisant diverses technologies selon le besoin.

Cependant certaines règles métier sont présentes dans le code source de l'application et de ce fait une modification de règle dans un langage ne porte pas la modification automatiquement dans un autre.

Ainsi, lors d'une modification qui touche potentiellement une autre technologie nous devons avoir une synchronisation à l'intérieur de l'équipe. Ce qui n'est pas forcément réalisable selon les situations et priorités de chacun.

L'absence de répartition des activités du groupe du point de vue informatique

Les effets de bord sont principalement présents lors de la modification de fonctions communes. Comme nous pouvons le voir sur la [Figure 4](#), celles-ci sont transversales à la suite logicielle et à toutes les activités du groupe.

Le problème qui se pose provient du manque de fiabilité lorsqu'une modification initialement prévue pour un logiciel se répercute dans un autre. Ce genre d'erreur est déjà difficilement acceptable lorsque cela impacte un logiciel distribué à l'intérieur du groupe mais pose encore plus de difficulté lorsqu'un client (utilisateur de GBM ou GRCLive) en est la cible.

Dans le contexte actuel (avec seulement deux clients externes) nous arrivons difficilement à tenir nos engagements et une qualité de service, cela justifie d'autant plus ce projet afin de pouvoir multiplier nos clients dans de bonnes conditions.

La multitude de base de données et leurs organisation

Plusieurs contraintes sont issues des bases de données et de la façon dont elles sont actuellement organisées.

Du point de vue technologique

À l'origine nous avons une base de données par technologie utilisée. Chacune embarque son propre système d'authentification et de gestion des droits. Bien entendu les données de production ne sont pas dupliquées en fonction de la technologie mais l'accès à celles-ci varient et sont donc sources d'erreurs.

Du point de vue des modules isolés

Certains modules sont également d'origine isolés dans une base de données distincte. Chaque logiciel communique donc avec plusieurs bases de données pour y accéder ce qui complexifie de surcroît l'accès aux données sans évoquer les coûts masqués en termes de performance.

Du point de vue des données stockées

Les bases rassemblent, des données qui n'ont pas le même niveau à l'intérieur du système d'information. Le mélange des données liées au fonctionnement du logiciel et de production pose des problèmes de synchronisation entre les logiciels.

⁸ Hypertext Preprocessor (PHP) est un langage de programmation libre utilisé pour produire des pages web dynamiques. C'est un langage interprété par un serveur web.

Du point de vue de la sécurité

L'accès aux données et le mode de communication entre les logiciels et les bases de données posent de plus en plus des problèmes de sécurité et de fiabilité des communications.

En effet, les logiciels (client lourd) accèdent directement aux bases de données pour effectuer des opérations. Ce qui veut dire qu'une personne mal intentionnée qui possède le logiciel (client lourd) a potentiellement accès aux données.

Le fait que chaque client soit autonome, possède des avantages et des inconvénients. Lors d'une période de stabilisation d'une application le fait que tous les utilisateurs ne soient pas perturbés par une opération qui pose problème est un point positif. Au contraire aujourd'hui nous avons besoin de fiabiliser les communications, pour se faire nous avons besoin de centraliser afin de maîtriser, les applications qui ont accès aux données.

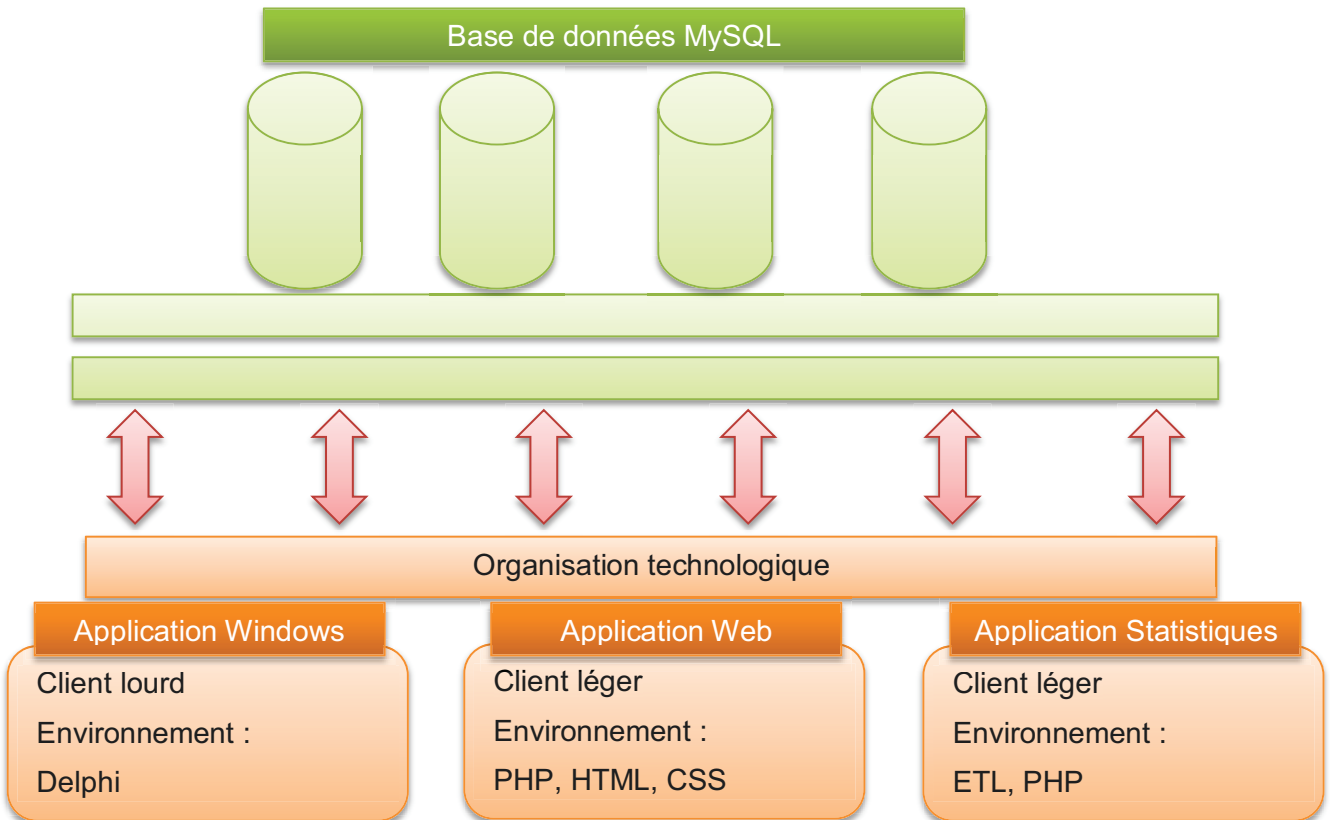


Figure 9 – Architecture des technologies existantes

3.2.1.2 Les critères de choix

Des objectifs et contraintes fonctionnels nous allons définir l'aspect technique et évolutif qui sera donc la référence pour la solution qui sera sélectionnée.

L'unification des fonctions communes et des règles métier sont la priorité des modifications de fond.

La multitude de langage implique que certaines fonctions qui devraient être communes sont développées plusieurs fois. Ce dernier point allonge considérablement la durée de maintenance car une modification doit être reportée à plusieurs emplacements. De plus la multitude de base de données est également un problème dans un objectif d'uniformisation. C'est pour cette raison que nous avons un système complexe dès sa base car il est difficile d'intervenir sur tous les systèmes de production.

Technologie robuste et éprouvée

La technologie choisie doit être suffisamment robuste et éprouvée pour assumer les objectifs. Ce qui met en évidence le fait que l'étape de recherche et développement est primordiale pour assurer le respect de cette contrainte.

Validation de l'environnement actuel

Cela vise à valider tous les systèmes qui gravitent autour d'un langage de programmation afin qu'ils soient compatibles. En effet, nous ne remettons pas en cause le système d'information dans son intégralité mais la structure logicielle. Cela constitue les contraintes environnementales à considérer comme des éléments qui ne sont pas affectés par les modifications attenantes au projet.

Ceci n'écarte pas la possibilité de moduler certains détails qui rendraient le système plus robuste et plus compatible. Nous traiterons individuellement chaque point lors de l'étude du cahier des charges technique afin de déterminer d'un commun accord avec les collaborateurs spécialisés ce qui peut entrer dans ce cadre sans apporter de nouveaux besoins au projet.

Appropriation par l'équipe de développeur

Le langage de programmation doit être utilisable par l'équipe de développement. L'utilisabilité est démontrée par deux principaux aspects :

- l'adaptabilité de l'équipe à une nouvelle plateforme de développement doit être possible sans formation externe à proprement dite, en effet aucun budget n'est dégagé pour assurer une acquisition de connaissance,
- la faculté de profiter des avantages de développement rapidement.

L'administration de la suite logicielle est pensée pour être rapide et réduite au minimum. Les règles métier et les configurations logicielles seront entièrement dynamiques par l'intermédiaire d'un système de liaison entre un client, une application et ses règles uniques. Cela représente une complexité certaine dans la conception, mais une fois ajustée, celui-ci peut faire gagner beaucoup de temps et de flexibilité.

Évolutifs

Ici nous retrouvons les principales contraintes issues directement de la direction commerciale du groupe. Cela consiste à avoir un produit unique permettant de considérer chaque logiciel, extranet ou autre comme un module.

Cela engage la possibilité de déploiement de module ou progiciels sur le même socle. Chaque demande sera analysée sous un angle générique afin d'être mis à disposition à d'autres clients du même secteur. Nous entendons en tant que configuration et règle métier tout ce qui peut être spécifique à un client afin d'avoir un code source unique, générique et dont la maintenance n'est pas liée à la connaissance accrue d'un secteur d'activité.

Nous distinguons donc les spécialités de chaque métier, afin de mettre en place une configuration type et de déployer les différences de façon générique. L'avantage est double, le client y voit un système puissant de modification et de prise en compte de ses règles métier. La maintenance applicative prend une tout autre dimension liée à la gestion d'une flotte applicative diverse.

3.3 Éléments de planning

Ce projet est né en plusieurs étapes. Dans un premier temps l'idée a germé entre moi-même et le Directeur Informatique d'Insiide ([Figure 5](#)) lors de réunions informelles régulières dans lesquelles nous réfléchissions à l'organisation du système d'information. Dans un second temps nous avons soumis nos idées à la direction qui évoque souvent une série de fonctionnalités attendues sur l'ensemble du groupe. De ces deux étapes ont débutés la réflexion globale sur les objectifs, [Figure 10](#), pendant laquelle la décision d'entreprendre ce projet a réellement était prise. Nous avons donc organisé le projet en jalon également présenté en [Figure 10](#) :

- choix technologique : recherche et développement,
- détermination des objectifs de produit Insiide,
- cahier des charges fonctionnel et technique,
- développement de la base du premier produit,
- consolidation du produit,
- mise à niveau des logiciels existants,
- recette, validation et mise en production.

Chaque repère marque une étape importante du projet et correspond au planning prévisionnel que nous avons définis au préalable. Seule l'étape de choix technologique, recherche et développement a été plus longue que prévue. En effet, ce choix est déterminant dans la continuité d'Insiide et oriente ses futures ambitions. Nous avons donc mis tous les atouts de notre côté afin d'assurer que ce choix n'allait pas être regrettable pour Insiide.

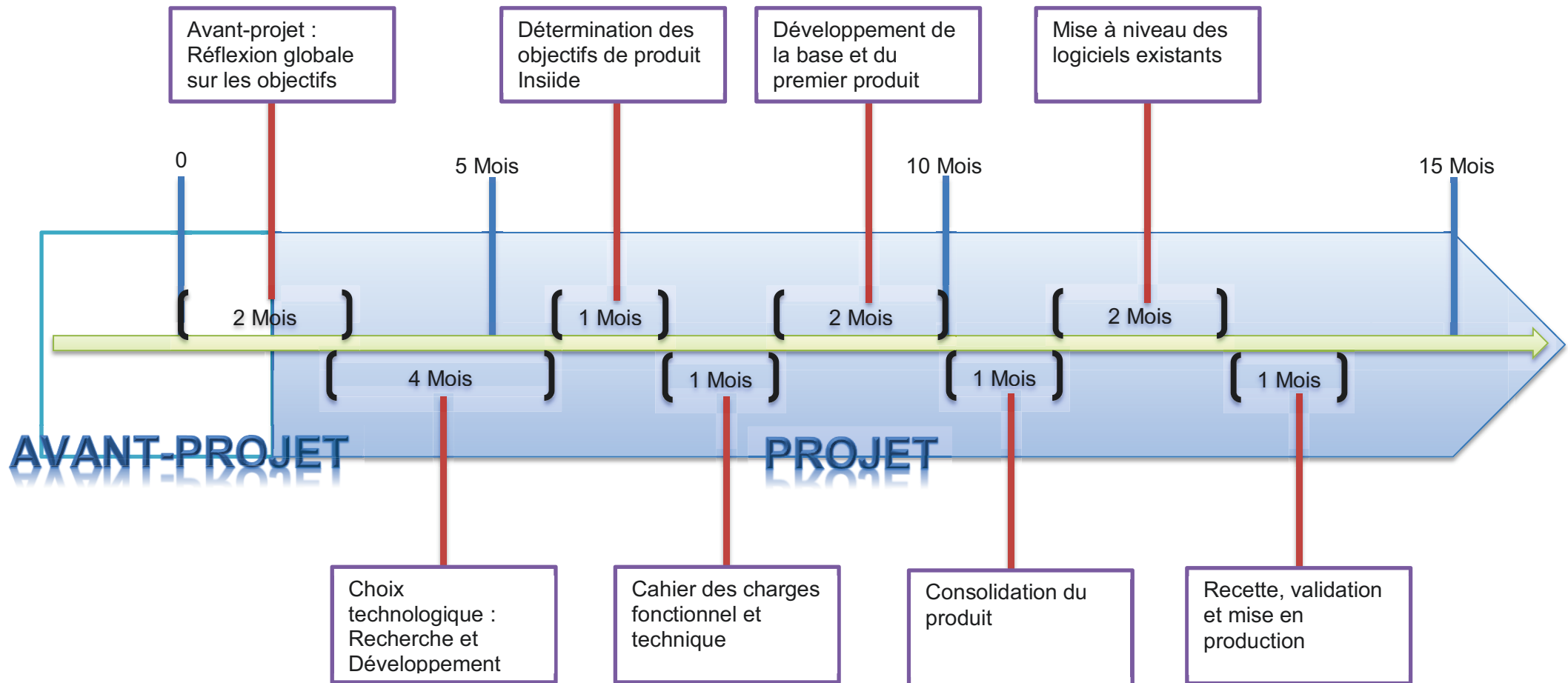


Figure 10 – Planification prévisionnelle du projet

La portée à terme de ce projet et sa réussite vont conduire Insiide à une rentabilisation de la structure. Aujourd'hui un des plus forts désirs est l'indépendance financière de celle-ci. À ce jour « Group Mayet Consultant » à un impact énorme sur les choix et décisions qui sont pris. Ceci se justifie par des sources de revenus générés par Insiide quasiment inexistantes. L'implication financière du groupe explique le penchant pour les fonctionnalités dédiées aux priorités internes. Les difficultés inhérentes à Insiide ne sont pas définies et perçues comme prioritaires et nécessaires au bon fonctionnement des outils informatiques avant ce projet.

Dès le début nous avons porté notre attention sur la démarche commerciale liée aux produits commercialisables. Les logiciels existants, sont d'une part des contraintes à respecter du point de vue fonctionnel, et d'autre part un exemple de savoir-faire interne sur la vision de l'informatique. Ils constituent donc un lourd historique à prendre en compte, puis une force démonstratrice d'innovation. Le système actuel nous affiche également ses limites qui vont être améliorées et contournées dans la nouvelle technologie.

La démarche commerciale consiste également à entrer dans un mode de gestion de projet jusque-là évité ou contourné pour parvenir plus rapidement aux objectifs. Le but est de mettre en place ce fonctionnement dont les vertus vont être curatives sur la progression de fond de la structure et de ses collaborateurs. Une rigueur différente sera nécessaire afin de collaborer avec les utilisateurs actuels et leurs responsables tout en mettant en place une planification et une priorisation des demandes.

Bien entendu le groupe restera toujours l'influent principal mais à terme des objectifs clairement fixés ne pourront être remis en cause au détriment de la gestion de projet mise en place par Insiide pour ces clients internes et externes. Concrètement cela veut dire qu'une société interne au groupe n'aura pas plus d'importance qu'une société externe, sauf si la priorité a été définie comme telle par la direction informatique en collaboration avec les responsables métier. Voici l'une des principales complexités de ce projet.

Cependant nous comptons sur les évolutions programmées et celles développées dans un objectif de rentabilité directe pour correspondre aux besoins de tous.

La planification et la répartition des temps passés ont forcément modifié voire perturbé l'organisation du service par rapport à l'intégration des nouvelles demandes. En effet, celui-ci était organisé sur des piliers de compétences et une répartition entre développeur. Aujourd'hui nous disposons d'une organisation plus flexible basée sur des fonctionnalités mutualisées à l'intérieur de l'équipe.

4. Évolution d'Insiide

Il convient pour atteindre les objectifs fixés de poser les problématiques d'ensemble et mettre en évidence les liens qu'elles peuvent avoir entre elles. Pour cela j'ai mis au centre du projet la concertation pluridisciplinaire, composé des principaux acteurs. Ces derniers se répartissent de la façon suivante, d'une part les collaborateurs du service informatique membres d'Insiide et d'autre part les responsables métiers représentant chaque entité. Cet ensemble est soigneusement encadré par la direction. Durant le projet une communication générale sera organisée afin d'informer l'ensemble des utilisateurs finaux des changements d'attributions et des personnes à contacter selon le sujet concerné.

4.1 La gestion de projet

Généralités

La gestion de projet ([Figure 8](#)) est la brique essentielle qui va assurer l'atteinte des objectifs tout en prenant en compte les contraintes et l'environnement. Le respect des échéances est également important afin que le comité de direction s'aperçoive de l'évolution correspondant à la planification définie.

L'avant-projet ([Figure 10](#)) correspond à l'étape de mise en place, de restructuration des modalités de communication, de remontée de demandes à Insiide et la clarification des moyens accordés à l'ensemble de ces modifications.

La désignation de responsable métier et la correspondance qui s'est créée est productive à tous les niveaux. Jusqu'à ce moment les utilisateurs n'avaient aucune visibilité sur le suivi des demandes. Désormais grâce à ces nouveaux intermédiaires nous allons avoir une action plus efficace.

En effet nous aurons désormais la possibilité :

- de mettre en commun chaque sujet, confronter les avis et les échanges permettront de comprendre l'aspect métier lié à chaque demande et d'apporter une réponse plus adaptée au besoin,
- de définir en toute transparence les priorités de chaque demande, évaluer et communiquer aux personnes concernées.
- d'avoir une logique et une cohérence de progression, le fruit d'une réflexion prenant en compte les sollicitations communes est garante d'un résultat qui convient à un maximum d'utilisateurs.

La réorganisation structurelle qui est effectuée ne peut pas se mettre en place en quelques jours. Il faut faire preuve de psychologie afin d'évoluer vers la ligne directrice établie.

Les responsables métiers

Pour synthétiser, les managers ont deux missions en amont et en aval des demandes. En amont, la responsabilité d'évaluer le bien-fondé d'une demande du point de vue fonctionnel. Si tel est le cas, ces derniers analyseront l'intégration et rédigeront un cahier des charges du besoin. En aval, ils établiront une recette finale après livraison d'une demande, et rédigeront une procédure fonctionnelle directement à destination des utilisateurs.

Ces missions ont directement pour but d'alléger la charge de travail du service informatique afin de dégager du temps aux nouvelles tâches qui sont nécessaires sur cette gestion de projet. En effet, il ne faut pas négliger le temps qui est nécessaire à faire vivre cette collaboration et les outils de suivi. De plus ce sont des durées qui sont difficilement quantifiables et qui étaient complètement absentes jusqu'à cette étape.

Uniformisation de la communication

Afin de cadrer au maximum cette collaboration des modèles de documents seront publiés par l'équipe informatique pour les échanges nécessaires. Du cahier des charges à la recette fonctionnelle les propriétés génériques seront évaluées et c'est sur une base unique que nous orienterons la communication. La gestion de projet détermine également les liens qui existent entre les demandes et constitue donc les jalons à l'intérieur d'un projet.

4.2 Recherche et développement

Je vais vous présenter la seconde étape de ce projet, elle permet de devancer bon nombre de problématiques et de prendre les décisions adéquates avant un engagement total.

Tout d'abord elle a un but de définition des attentes techniques, de la technologie et du langage de programmation qui va être utilisé. Il y a énormément d'objectifs qui sont liés à cette partie et aux procédés techniques qui vont être choisis.

Cette démarche de recherche et développement se situe après la mise en place de la gestion de projet pour deux raisons. La première est la faculté à canaliser les demandes ce qui est essentiel pour éviter les perturbations. La seconde consiste à planifier rigoureusement les objectifs qui sont testés mais aussi de récolter les informations et les exploiter pour la suite du projet. En définitive ce sont les outils de gestion de projet qui assure le respect des contraintes mais également un suivi de l'évolution.

Cette phase d'expérimentation va donc déterminer que l'ensemble des attentes techniques ont bien une solution. Celles-ci sont déterminées en fonction de la connaissance que nous avons des logiciels actuels afin d'en dégager les besoins. Nous y ajoutons les spécificités attendues ou les évolutions que nous souhaitons réaliser, d'un point de vue fonctionnel.

Nous nous assurons que l'ensemble des objectifs et contraintes listés sont bien représentés et vont bien être validés par une solution technique.

Concrètement la R&D s'est invitée dans les réunions d'avant-projet lorsque nous avons besoin pour concevoir le projet d'avoir des certitudes ou des hypothèses fiables.

Parmi les solutions, les avantages et inconvénients seront envisagés et l'ensemble des informations permettront de faire un choix par la suite.

Cette phase est également transversale durant le projet si le besoin de valider certains éléments se ressent. Elle permet également d'appréhender le budget nécessaire à cette nouvelle technologie et de le faire valider à la direction.

4.3 Définition des cahiers des charges

Comme précédemment cité, le projet peut se scinder en deux grands axes. La consolidation et l'évolution technologique pour les solutions logicielles existantes et le développement de nouveaux produits qui seront ajoutés au catalogue produit d'Insiide.

Chacun des axes est porté par une partie de l'équipe pour avancer simultanément.

Le premier axe, est sous ma responsabilité en termes de définition du cahier des charges fonctionnel et technique. Comme je l'ai déjà évoqué, du point de vue fonctionnel, nous reproduisons les fonctions communes des applications existantes. L'essentiel est de n'induire aucune perturbation du point de vue de l'utilisateur. Nous avons pris le temps de poser clairement les différentes parties que nous allons migrées et de définir les fonctionnalités afin que ce soit le plus clair possible pour l'ensemble de l'équipe en cas de reprise du projet par un autre développeur par exemple. J'ai tenu à ce que l'ensemble des informations soient posées avant de débiter le développement. Cela est difficile à tenir lorsque des pressions ou des priorités s'exercent sur l'équipe.

Le second axe est porté par les deux personnes qui nous ont rejoints en partenariat au sein d'Insiide depuis le début du projet. Ce sont les responsables métier d'un projet baptisé « Association », qui est un logiciel de gestion pour association sportive spécialisé dans le Rugby. Ils sont donc conviés aux réunions hebdomadaires du comité et demandent de porter à l'ordre du jour les éléments qui leur paraissent utiles pour le développement de ce produit.

En termes de planification, nous pouvons voir que les cahiers des charges ont débutés qu'à partir du 6^{ème} et 7^{ème} mois de projet. Cette planification montre de quelle façon nous l'avons intégré du point de vue d'Insiide. Les deux personnes qui nous ont rejoints, ont, eux, débuté le cahier des charges fonctionnel avant même la définition de notre partenariat, c'est-à-dire avant même le début du projet. C'est en effet un logiciel à part entière qui nécessite d'être entièrement construit. Ce qui est intéressant c'est effectivement que nous avons mis en commun nos objectifs et contraintes à partir du 6^{ème} mois afin d'ajuster conjointement le cahier des charges pour correspondre aux exigences de chacun.

Cette mise en commun n'a pas été évidente car ce sont deux mondes différents qui devaient communiquer pour mener le même projet de fond. L'évolution du projet est garantie par les comités organisés ou chaque partie est représentée, afin d'éviter toute confusion possible.

4.3.1 Fonctionnel

4.3.1.1 Suite logicielle existante

Sur la suite logicielle existante nous avons un fil conducteur central qui consiste à réduire d'une part les temps de développement et d'autre part d'assurer une administration logicielle commune à tous les produits proposés par Insiide. Pour cela nous avons deux axes de progression car nous devons toujours prendre en compte les besoins du groupe encore très influant dans l'avancée globale d'Insiide. Nous devons donc trouver un compromis entre :

- améliorer le processus d'administration des applicatifs maintenus,
- engager les nécessités les plus fortes vers une réalisation.

Dans un premier temps mon rôle a été de définir quelle sera notre console d'administration. Celle-ci est utile à plusieurs fins :

- gestion des règles métier sans modification de code source,
- diminution sensible des temps de maintenance,
- délégation à certains responsables des règles de gestion de leur activité.

Plus globalement nous avons réfléchi à une mutualisation des fonctions communes. Comme nous l'avons vu sur le schéma, [Figure 4](#), bien qu'il y ait des fonctions dites communes, celles-ci ne sont pas vraiment mutualisées lorsqu'elles sont dupliquées sous plusieurs langages de programmation. Voici donc notre premier chantier, définir les fonctions qui doivent être communes quel que soit l'environnement de développement.

Ensuite j'ai isolé les nouvelles fonctionnalités qui seront prioritairement développées dans la nouvelle technologie afin de définir le besoin et les attentes en termes de résultat. J'ai pour cela rédigé, d'un commun accord avec les responsables métier concernés, des cahiers des charges pour chacune de ces parties. Ce qui permet, une fois de plus, d'assurer l'intégration d'un développeur à n'importe quel moment du projet, de manière plus efficace et de les inclure dans une planification au plus juste.

Par ailleurs, toute notre attention s'est concentrée sur une base qui va nous permettre d'ajouter des modules fonctionnels progressivement. De nouvelles fonctionnalités pourront être ainsi développées, qu'elles soient directement planifiées ou exécutées par la suite.

Nous avons donc mis au centre de nos préoccupations une nouvelle notion d'application disponible à un client. C'est une notion complexe que nous avons définis pour répondre à l'ensemble de nos problématiques. Nous entendons par client, tous les acteurs à qui nous fournissons un logiciel, y compris les entreprises du groupe pour qui nous introduisons cette notion de client. Concrètement nous ne facturons pas en interne alors qu'Insiide n'est justement pas autonome et rien ne lui permet de le prétendre. Par contre, cela clarifie la situation, plutôt que de faire des solutions pour des acteurs externes et d'autres en interne, ce sont des solutions pour tous les clients d'Insiide.

Nous nous sommes également axés sur une notion centrale et une fonction commune à l'ensemble des applicatifs, la mutualisation des utilisateurs, de leurs droits, et de leurs rôles.

Ceci est typiquement un projet de fond que nous n'avons jamais pu mener dans les anciennes technologies. Nous sommes pourtant persuadés que cette restructuration est la clé d'un allègement considérable des temps de développement et de maintenance.

Ainsi du point de vue fonctionnel hormis les fonctionnalités qui ont été définies comme prioritaires nous avons surtout des objectifs globaux qui sont propres à la gestion du système d'information et qui concerne donc uniquement l'équipe informatique du groupe. Nous avons considéré que c'est un point clé à la réussite de ce projet.

Il s'est avéré compliqué d'expliquer précisément, à la direction notamment, les véritables avantages à engager ce chantier. Cependant les temps de réflexion et de réalisation ont démontré qu'il était nécessaire de clarifier nos besoins qui n'ont jamais pu s'avérer prioritaire par rapport aux demandes du groupe.

Si l'on devait résumer, l'objectif fonctionnel de notre projet consiste à créer une base commune à tous les applicatifs sans perturber les utilisateurs finaux des plateformes de production. Une fois ce nouveau socle commun posé et fonctionnel, d'autres intégrations de module pourront être effectués.

4.3.1.2 Nouveau produit

Pour l'instant nous avons un seul nouveau produit qui est intégré au projet. Celui-ci n'est pas une fin en soi et ne va pas permettre à Insiide d'être autonome financièrement. Par contre c'est un produit qui multiplie nos clients directs à qui nous pourrions progressivement proposer des services additionnels de maintenance ou de produits décisionnels par exemple.

Ce produit c'est également une façon de débiter l'ouverture tout en développant un produit assez simple fonctionnellement parlant. En effet, en termes de ressource nous ne pouvons pas mener une édition de logiciel conséquente en même temps que nos restructurations technologiques.

Par contre, il est vrai que c'est un cœur de métier qui est complètement inconnu de l'équipe et avec lequel nous devons nous accoutumer. Les ressources externes qui portent plus précisément ce projet sont aussi là pour traduire les besoins en fonctionnalité intelligible pour Insiide.

L'édition de celui-ci s'est programmée en trois étapes :

- mise en commun et adaptation du cahier des charges fonctionnel,
- rédaction du cahier des charges techniques,
- développement du logiciel en mutualisant les fonctions communes.

4.3.2 Technique

A ce stade nous avons définis les objectifs et contraintes ainsi que les attentes fonctionnelles du projet. Il reste donc à déterminer la structure technique de celui-ci.

Celle-ci doit permettre de concrétiser les éléments suivants :

- mutualisation des fonctions communes,
- application client/serveur,
- optimisation des bases de données,
- approche modulaire des fonctionnalités,
- intégrer la notion de client sur les logiciels proposés.

Le cahier des charges technique repose sur la recherche et développement qui a été menée afin de concevoir le système qui répond aux attentes.

Nous avons deux types de caractéristiques à traiter :

- les nouvelles fonctionnalités qui entraînent des impacts sur la suite logicielle existante,
- celles qui ont pour but de corriger les problématiques actuelles.

Et dans le même temps deux niveaux de demandes :

- refonte des fonctionnalités communes,
- développement de nouveau produit ou fonctionnalités.

Alors que nous avons réalisé un cahier des charges fonctionnel pour chaque évolution intégrée dans ce projet, nous concevons un seul cahier des charges technique qui répond à l'ensemble.

Par rapport au délai d'un mois qui est annoncé dans la planification ([Figure 10](#)) il est évident que celui-ci ne peut intégrer une réflexion complète sur une refonte de cette envergure. Insiide a anticipé les attentes notamment, lorsque j'ai dans le cadre de mes missions, entamé une démarche de veille technologique pour améliorer le système d'information dans sa globalité. Ce projet en est une concrétisation qui avait besoin d'être réalisée.

4.4.1 Choix de la plateforme de développement

Pour retenir une solution nous avons mis en concurrence deux technologies pour répondre à nos exigences. Cette mise à niveau a également pour but de choisir une plateforme récente, en pleine évolution, mais aussi robuste et éprouvée sur des systèmes exigeants.

Nous avons opposé l'environnement JAVA et dotNET en comparant les mêmes fonctionnalités et les résultats obtenus. J'avais à ce moment-là approximativement le même niveau d'expertise dans les deux langages de programmation, que je situerai de moyen. Le but est également d'évaluer le temps nécessaire au développement ou à la prise en main selon un niveau établi.

En effet, deux paramètres sont essentiels dans la programmation :

L'équipe, relativement petite et qui de ce fait, ne peut se permettre d'éprouver des difficultés de conception.

La prise en compte d'une phase d'adaptation qui doit être estimée dans les délais. Cette phase correspond à l'intégration de l'environnement technique afin de réaliser des tests à but d'apprentissage. Cela nous permet de nous accorder sur les mêmes règles adéquates à appliquer par toute l'équipe lors de la réalisation.

L'environnement de développement historique dans le groupe est le Delphi pour les solutions applicatives et les langages HTML et PHP pour les solutions web. Chaque membre de l'équipe possède des compétences dans les langages utilisés avec certaines spécialités ou affinités.

Le développement dans le respect des standards et des règles établies est au cœur de la réalisation et ne doit pas être négligé. La mise à jour des connaissances et la documentation a été constante durant cette étape. En effet, nous désirons s'engager vers de nouveaux horizons, tout en construisant une base solide qui pourra tenir la charge et évoluer.

Comme je l'évoquai, avec la taille de l'équipe, nous ne pouvons pas nous permettre de produire une application sans aucune aide extérieure. La disponibilité de composants plus ou moins avancés, d'une communauté active, d'un environnement professionnel est indispensable à la réussite.

J'ai organisé la comparaison en deux parties :

- la partie cliente qui comprend les interfaces graphiques, les interactions avec l'utilisateur et le système,
- la partie serveur, qui regroupe les méthodes qui peuvent être utilisées par le client et qui assurent la communication avec les bases de données.

Cette distinction nous permet d'être plus efficaces dans l'étude des résultats et assure que les critères étudiés soient identiques quel que soit la technologie.

Une des limites que nous retirons de l'environnement Delphi est la multiplication des composants et des interactions systèmes pour servir telle ou telle fonctionnalité. En finalité ce sont autant de sources de problème que l'on ne maîtrise pas et auxquelles nous sommes liés pour évoluer. C'est pour cette raison qu'il faut évaluer à chaque fois les avantages et les inconvénients lors de l'intégration d'une suite de composant. Par conséquent, nous choisirons et utiliserons une seule suite assez complète, qui répond au mieux à un éventail de possibilités.

Par ailleurs, la technologie choisie va déterminer l'environnement structurel et serveur nécessaire, afin d'évaluer notamment les coûts de ce dernier.

4.4.2 Analyse et résultat de l'étude

Nous avons pris en compte point par point les différents éléments qui nous permettent de faire un choix. Le résultat de cette étude révèle que les principales applications doivent utiliser une technologie client/serveur afin de répondre aux attentes tout en respectant les problématiques. En effet, la [Figure 9](#) nous présente les applications en communication directe avec les bases de données. Cela est perfectible pour les raisons suivantes :

- Le client doit gérer la multitude de base de données
 - connexion aux différentes sources d'information,
 - savoir où est l'information quand celle-ci se trouve dans une autre source.
- Problème de sécurité des données qui transitent par internet
 - l'applicatif s'authentifie aux bases de données,
 - mouvement des informations transmises du logiciel aux bases de données et inversement.
- Mise à jour
 - lors de la modification de règle métier il faut mettre à jour l'ensemble des applications,
 - les mises à jour ne sont pas unifiées et il faut un système complexe pour les gérer.
- Pas de centralisation du code source multi technologies
 - l'aspect métier est dupliqué autant de fois que de langage de programmation utilisé,
 - les algorithmes entre deux technologies doivent toujours être synchronisés afin que les mêmes informations soient disponibles.

Pour conclure le mode de communication client/serveur ([Figure 11](#)) est exploitable pour ses principaux avantages :

- centralisation sur le serveur de tous les accès aux sources de données,
- facilité de mise à jour et mise en place de contrôle de sécurité,
- toute la complexité est reportée sur le serveur, le client peut être simplifié au maximum,

Il existe malgré tout, des inconvénients à cette infrastructure :

- les serveurs doivent être capables de répondre à une volumétrie de requête élevée,
- si le serveur ne fonctionne plus, plus aucun client ne fonctionne,
- coût de mise en place et de maintenance.

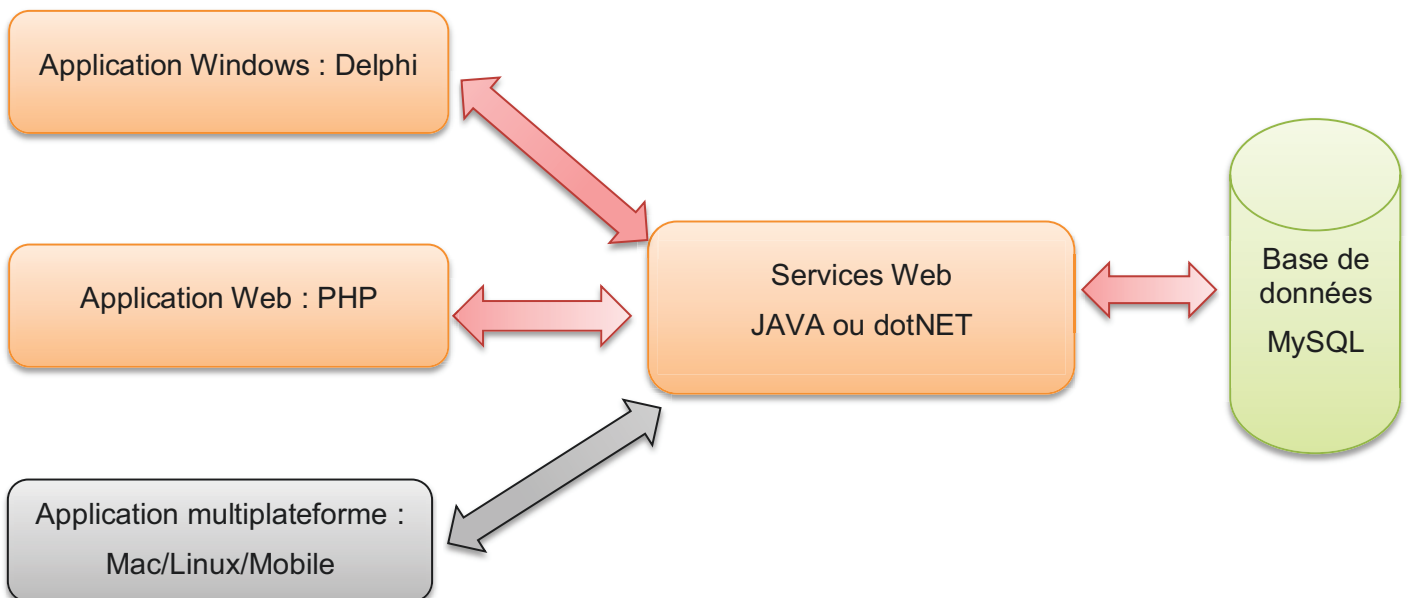


Figure 11 – Architecture schématique d'une solution client/serveur



Au vue des différentes informations, le JAVA est, selon moi, la technologie de base (en tant que serveur) qui répond à l'ensemble des demandes. Celui-ci se fonde sur les constatations suivantes :

- l'environnement serveur actuel, Linux à cent pourcent, est déjà prêt pour accueillir l'hébergement d'application JAVA, à contrario du dotNet qui nécessite un environnement Microsoft Windows,
- l'environnement Microsoft modifierait d'une part notre paysage serveur (et remet donc en question d'autres services actuel), d'autre part nous contraint à une rigueur budgétaire plus élevée,
- l'équipe de développeur n'a pas plus d'affinité ou de compétence avec l'une ou l'autre des technologies,
- l'édition d'application multi plateforme, y compris mobile est facilitée

Par ailleurs des limites sont décelées pour cet environnement :

- difficultés pour créer rapidement des applications clientes esthétiques,
- interaction système ou suite bureautique non nativement supportée.

Les inconvénients afférents à l'infrastructure client/serveur sont donc maîtrisés par cet environnement.

Un point identique quel que soit la technologie choisie est effectivement la maintenance de la solution. Sur ce point, la comparaison entre la maintenance d'une flotte de client et d'une flotte de serveur (plus réduite forcément), ne révèle pas de différence problématique pour le service.

Le socle serveur peut être utilisé par n'importe quelle autre technologie.

Les applications Windows resteraient développées en Delphi (pour la partie interface) et utiliseraient les requêtes publiées par le serveur.

Les applications WEB développées en PHP, consomme les services web au lieu de faire directement appel aux bases de données.

Ainsi les clients riches (Delphi) et clients légers (PHP) feraient appel aux mêmes fonctions serveur, aux mêmes règles métier. De plus, la possibilité et l'augmentation des plateformes qui supportent le JAVA étant de plus en plus nombreuses cela nous permet à l'avenir d'intégrer des applications sur ce type de plateforme plus facilement.

La décision finale du choix de technologie revient au Directeur Informatique, correspond à la technologie Microsoft dotNet pour le développement d'application au sens large du terme.

Le dotNet a été testé au même titre que le JAVA et répond à l'ensemble des attentes techniques. Les différences les plus marqués sont :

- qu'il représente un effort d'investissement immédiat sur l'environnement serveur,
- qu'il ne permet pas d'ouvrir la possibilité d'édition de logiciel multiplateforme.

L'idée, ne consiste pas à éditer des logiciels sur les plateformes spécifiques telles que le permet le JAVA, mais de créer des solutions web en dotNet accessibles depuis n'importe quelle plateforme qui inclut un navigateur web. Même si l'interactivité est différente, on tend à réduire les différences qu'il peut y avoir entre une application web et une application développée spécifiquement.

Une fois le choix technique prit, et validé symboliquement par la direction du groupe, nous avons poursuivi les différentes étapes techniques de la réalisation. Symboliquement, car le choix technologique reste tout de même un choix important pour une entreprise, mais cela insiste sur le fait que nous sommes entièrement responsable du système choisi.

5. Réalisation technique

Nous allons mettre en évidence le contexte dans lequel la réalisation a été effectuée, autrement dit les ressources qui vont nous permettre d'atteindre les objectifs fixés tout en respectant les contraintes.

La réalisation a été l'étape la plus longuement préparée, à juste titre car une évolution technologique de cette envergure doit être réfléchi. D'autant plus que celle-ci est liée à une réorganisation structurelle qui fait partie pour moitié de la réalisation.

Les applications de la suite logicielle ([Figure 4](#)), sont utilisées quotidiennement par 400 à 500 utilisateurs environ. Nous avons donc concentré nos efforts afin d'intervenir sur les plateformes de production sans faire subir de désagrément.

Le retour d'expérience tout comme la communication pendant le projet reste une de mes principales préoccupations car je suis conscient que celui-ci permettra d'isoler les limites qui seront à surmonter.

5.1 Avant le développement

Le début du développement et les premières livraisons transversales sont très attendues lorsque le projet a débuté depuis 5 à 6 mois. Cependant il est facile de se précipiter, de partir sur de mauvaises bases et d'avoir par la suite à un système difficile à maintenir. C'est justement ce que nous souhaitons corriger avec cette évolution technologique.

J'ai extrapolé ma vision à tout ce qui peut représenter un obstacle pour Insiide à court, moyen et long terme. Il s'avère que l'aspect juridique ou légal de l'édition de logiciel est très important d'autant plus lorsque l'activité principale consiste à éditer et distribuer des logiciels sur le marché.

5.1.1 Ordonnancement des fonctionnalités

Chacune des parties est liée à une ou plusieurs fonctionnalités. Nous les avons ordonnancées et planifiées dans l'outil de gestion de projet afin d'établir des mises en communs avec les collaborateurs d'Insiide et des livraisons transversales. Le partage de l'information est important dans la démarche car il laisse la possibilité à l'équipe de s'exprimer sur le système à maintenir dans un futur proche.

J'ai été affecté à plusieurs types de rôle afin d'aboutir au développement du serveur d'application en dotNet.

Rôle de développement

Lors de la planification j'ai pris la responsabilité de débiter le développement d'une page blanche pour créer un socle qui guide au maximum le développeur. En effet, lors de la conception, j'ai fait en sorte d'assurer une qualité du code source, commenté, documenté en respectant au maximum les standards sauf s'ils sont vraiment trop contraignant pour une petite équipe.

Rôle de référent technique

À l'intérieur de l'équipe mais aussi en rapport avec les responsables métier ou la direction.

En communication avec les développeurs d'Insiide j'ai fait en sorte d'être un maximum disponible surtout au début. D'une part afin de respecter la planification prévisionnelle à laquelle je tenais tout particulièrement quant à mon engagement. D'autre part, je suis à l'origine de cette évolution technologique, et j'ai à ce titre un devoir moral envers mes collègues afin de les épauler un maximum. La bonne entente à l'intérieur de l'équipe est un moteur incroyable et je pense qu'elle est un des ingrédients de la réussite que nous avons connu.

En communications avec les responsables métier ou la direction, mon rôle est différent. Il consiste essentiellement à faire comprendre les raisons des décisions d'une façon intelligible pour tous. C'est-à-dire dénuées de technicité. J'ai souhaité être transparent sur toutes nos actions, car nous avons tous le même objectif à l'intérieur du groupe et cela aurait créé des tensions inutiles.

Durant le projet j'ai été l'interface entre l'équipe informatique et le reste du groupe, étendue à l'ensemble des activités de ce dernier. C'est bien sûr une lourde tâche d'être l'interlocuteur potentiel de toutes les personnes qui peuvent avoir des demandes. Toute la complexité se situe dans l'analyse de la demande et dans la priorisation en fonction de tous les paramètres. Parfois la décision doit intervenir dans des temps restreints car elle est pour le demandeur d'une priorité élevée.

5.1.2 Outils et plateforme de développement

Une nouvelle technologie implique l'acquisition et l'appropriation de nouvel environnement de développement par les développeurs. La solution qui a été sélectionnée, le Microsoft dotNet implique un environnement de conception et de déploiement.

Acquisition d'un environnement

Dans un objectif de maîtrise des coûts j'ai opté pour une solution commercialisée par Microsoft spécialement intéressante pour les structures dans notre cas. Sous réserve de respecter les conditions requises nous pouvons souscrire à un pack qui se nomme « Microsoft Action Pack⁹ », désormais « Microsoft Action Pack Development and Design » (MAPDD). Afin d'utiliser ce pack il y a trois étapes préalables :

- rejoindre le réseau « Microsoft Partner Network¹⁰ »,
- réaliser une évaluation,
- commander l'abonnement.

Ces étapes sont conçues pour limiter la commande aux entreprises de moins de cent employés ayant globalement une activité de développement ou d'intégration de solution informatique.

Nous avons déterminé qu'Insiide respecte les conditions minimale requises et peut donc profiter du contenu de ce pack et plus particulièrement de :

- Visual Studio 2010¹¹, environnement de développement,
- Windows Server 2008¹², environnement serveur d'hébergement d'applications,

Globalement pour trois développeurs si nous n'étions pas passés par le MAPDD l'investissement aurait été presque sept fois supérieur au prix de l'abonnement. C'est donc une solution qui nous permet de débiter le développement dans une cohérence budgétaire raisonnable par rapport aux ressources de l'entreprise.

J'ai accordé de l'importance à cette partie car nous avons tendance à sous-estimer le poids que cela peut représenter et les répercussions que cela peut engendrer. En effet, dans un contexte déjà difficile, alourdir considérablement l'investissement de départ n'est pas viable et aurait pu entraîner l'annulation de ce projet. Dans le meilleur des cas, cet investissement aurait été une source de pression supplémentaire dans la réalisation.

Ce dernier point met en évidence que pour s'engager dans une nouvelle technologie nous devons avoir un minimum de connaissance de cet environnement car durant l'évolution de projet on peut facilement être pris de cours en termes de temps lorsque nous devons systématiquement rechercher une information.

Heureusement pour Insiide et pour le projet, mon expérience dans les deux solutions technologique étudiées m'a permis d'être très réactif sur l'ensemble de ces sujets. J'ai d'ailleurs été fort de proposition pour appréhender un maximum de difficulté. J'ai également pu porter un sentiment de sécurité et de maîtrise de cette évolution technologique afin de rassurer tout particulièrement la direction du groupe.

⁹ Pack de licence souscrit auprès de Microsoft par le biais d'un abonnement qui donne droit à l'utilisation à des fins de tests et de développements certains produits.

¹⁰ Réseau de partenaire de Microsoft qui donne droit à un certain nombre d'avantages, comme être référencé parmi les partenaires officiels. Il existe plusieurs niveaux de partenariat.

¹¹ Visual Studio 2010 est un ensemble complet d'outils de développement pour la plateforme dotNET.

¹² Windows Server 2008 est un système d'exploitation édité par Microsoft orienté serveur.

La plateforme de développement

La plateforme de développement est composée de deux parties :

- le poste de développement,
- l'environnement serveur de compilation et de test des applications.

Le poste de développement peut être perçu comme un détail mais il doit être facilement installé, configuré et maintenu pour être opérationnel. En effet, j'ai l'expérience de préparation fastidieuse dans les technologies actuelles autour de Delphi, car nous avons le RAD Delphi, puis la quantité de composants additionnels à installer et à configurer. De plus, cela est une compétence peu présente chez un technicien système qui aurait la charge de déploiement de poste de travail. J'ai considéré que l'installation et la maintenance du poste de développement devait être revue car elle représente une charge de travail réelle qui n'est présente dans aucune planification mais qui ralentit le développeur au quotidien. C'est également pour cette raison que j'ai limité à une seule suite de composant qui répond à un éventail de possibilités.

L'environnement serveur a été pensé pour assurer une qualité de conception des applications et une rapidité d'utilisation pour le développeur. Toujours par retour d'expérience par rapport au fonctionnement actuel du service je me suis posé la question sur la façon d'améliorer et de résoudre les problèmes rencontrés lors des processus suivant :

- compilation des applications¹³,
- le déploiement des applications, c'est-à-dire la mise à disposition aux utilisateurs,
- test des applications sur une plateforme représentative de la plateforme cible.

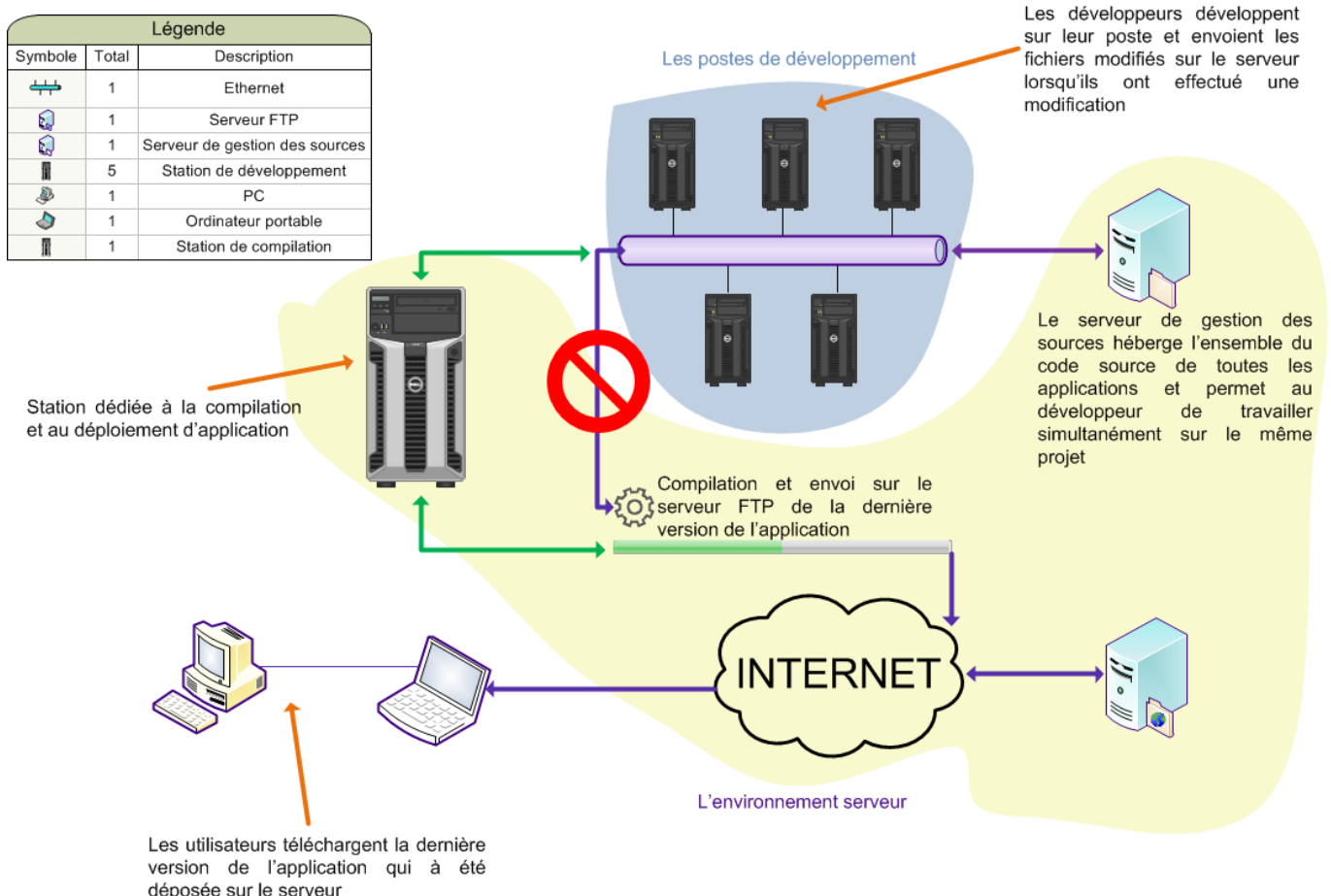


Figure 12 – Évolution du déploiement d'application

¹³ La compilation d'une application consiste à transformer le code source écrit dans un langage de programmation en un autre langage informatique qui puisse être exécuté sur une plateforme cible (le poste de l'utilisateur).

Le déploiement d'application

L'analyse de la [Figure 12](#) nous montre la principale modification que je souhaite apporter au fonctionnement du service lors du déploiement d'application : une station dédiée à la compilation et au déploiement d'application. En effet, j'ai identifié une source de problème lorsque chaque développeur peut indépendamment créer une nouvelle version d'un logiciel.

Cela est fondé sur les constatations suivantes :

- les développeurs ont des configurations spécifiques qui peuvent différer entre environnements de test et de production,
- avec la multitude de composants utilisés sur l'environnement Delphi il arrive qu'un composant soit mal installé ou ait été dégradé sur un poste, dans ce cas l'utilisateur final télécharge une application instable,
- il arrive que le développeur veuille publier une nouvelle version de logiciel alors que sur son poste il a une version du code source qui comporte des modifications qui ne doivent pas être compilés dans une version de production.

J'ai donc pris la décision de soumettre l'idée au Directeur Informatique afin de corriger ce point qui fait défaut dans le service. En effet, les défauts qui peuvent émerger de cette situation, sont une charge de travail supplémentaire et une source de problème que l'on peut pourtant maîtriser en amont.

J'ai à cette occasion apporté une compétence à Insiide en déployant un système serveur dit hyperviseur¹⁴, capable d'héberger des machines virtuelles¹⁵. Le système VMware ESXI¹⁶ nous permet de profiter d'une station de compilation et de déploiement accessible au développeur. Ainsi nous contrôlons les coûts afférant au développement d'application en hébergeant plusieurs machines qui ont des rôles différents sur le même serveur. Mon idée consiste également à créer une plateforme virtuelle de test d'application avant la mise en production et le déploiement d'une nouvelle version.

La seconde modification apportée, correspond à l'uniformisation du système de gestion des sources. Celui-ci héberge la dernière version du code source des applications et la partage à l'ensemble de l'équipe. Nous utilisons deux systèmes différents selon le langage de programmation utilisé, le système Starteam¹⁷ pour les applications Delphi et Subversion (SVN¹⁸). Nous avons décidé de mutualiser le système de gestion des sources en l'occurrence SVN quel que soit l'environnement de développement.

La réorganisation nécessaire

La nouvelle machine créée pour compiler et déployer les applications est accompagnée d'une réorganisation du service informatique. J'ai été nommé responsable, afin d'apporter de la rigueur dans la création de nouvelles versions logicielles. Pour cela j'ai mis en place quelques règles qui me semblent importantes pour assurer une qualité de service excepté cas de force majeure :

- décrire la raison pour laquelle un développeur modifie un fichier lorsqu'il le stocke sur le serveur de gestion des sources,
- seul le responsable effectue des versions, trace l'historique et gère la notion de version majeure et mineure de chaque application,
- un cycle de test de l'application est impérativement effectué avant tout déploiement d'une nouvelle version.

Il est vrai que ce processus est un peu lourd pour un développeur qui peut parfois effectuer des nouvelles versions en urgence. C'est également pour cette raison que personne ne souhaitait (hormis moi-même) prendre cette responsabilité.

Étant donné que nous allons avoir un système hybride pendant quelques temps, lorsque nous allons avoir une cohabitation des applications Delphi, PHP et dotNet, il est fortement recommandé d'effectuer ces évolutions. La plupart avaient été appréhendés durant la période de recherche et développement et de choix de la technologie retenue.

¹⁴ Plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une machine physique en même temps.

¹⁵ Une machine virtuelle est à l'intérieur d'un hyperviseur, une reproduction de composant d'un appareil informatique afin de faire fonctionner le système d'exploitation classique.

¹⁶ C'est un hyperviseur édité par la société VMware, il en existe d'autres.

¹⁷ Système de gestion de versions de code source d'une application, édité par Borland.

¹⁸ Système de gestion de versions édité par Apache et BSD, édité par CollabNet.

L'organisation du service s'est mise en place progressivement, mais le gain pour le développeur est si important que je n'ai pas eu de mal à argumenter cette nouvelle méthode de travail.

Dans ces opérations nous allons faire en sorte de modifier la fréquence entre deux nouvelles publications d'application. Nous avons constaté que celles-ci sont trop nombreuses et pas suffisamment encadrées. L'objectif fixé est de réduire le nombre de version à une par semaine, puis une par mois en fin de projet. Ce temps doit nous permettre de traiter un certains nombres de demandes entre deux versions et d'assurer une cohérence dans les développements.

5.2 Développement et consolidation

Le développement a débuté comme nous pouvons le voir sur la [Figure 10](#) à l'issu des trois premières étapes :

- choix technologique : recherche et développement,
- détermination des objectifs de produit Insiide,
- cahier des charges fonctionnel et technique.

Il est lui-même composée de trois parties que je vais vous détailler :

- développement de la structure applicative,
- consolidation du produit,
- mise à niveau des logiciels existants.

5.2.1 Développement de la structure applicative

La nouvelle architecture technologique est le socle de la base du premier produit ainsi que la mise à niveau de la suite logicielle existante. Celle-ci présentée en [Figure 11](#), consiste principalement à héberger sur un serveur d'application des services web utilisés par les applications clientes (qu'elles soient client riche ou client léger). Nous appelons cela une infrastructure client/serveur, car les clients adressent leurs requêtes au serveur d'application plutôt que de contacter directement la base de données.

Cependant pendant la période de mise en place qui se veut progressive, les applicatifs clients existants interrogerons toujours les bases de données (représenté par le système de transition [Figure 13](#)). Ces dernières seront donc toujours sollicitées par plusieurs types de technologie et à différent niveau de l'architecture. Comme nous pouvons le constater le but est de déporter la logique applicative de la partie cliente vers la partie serveur, seule la partie présentation reste sur la partie cliente.

Ainsi en plus d'une application client/serveur nous pouvons parler d'une application « n-tier¹⁹ » où la logique applicative ([Figure 13](#)) est divisée en plusieurs couches²⁰ :

- couche présentation,
- couche des services web,
- couche métier,
- couche d'accès aux données.

Ainsi comme nous pouvons le voir sur la [Figure 13](#) dans le déplacement de la logique applicative nous nous ne déplaçons pas la couche « présentation » qui reste sur la partie cliente.

¹⁹ Architecture n-tier : architecture qui consiste à séparer en différentes couches les différentes parties qui constitue une application. Cette logique consiste à donner un ordre aux couches et où le développeur s'interdit à invoquer une couche qui n'est pas directement inférieure ou supérieure.

²⁰ Couche (applicative) : chaque couche communiquent entre elles pour proposer un ensemble de service.

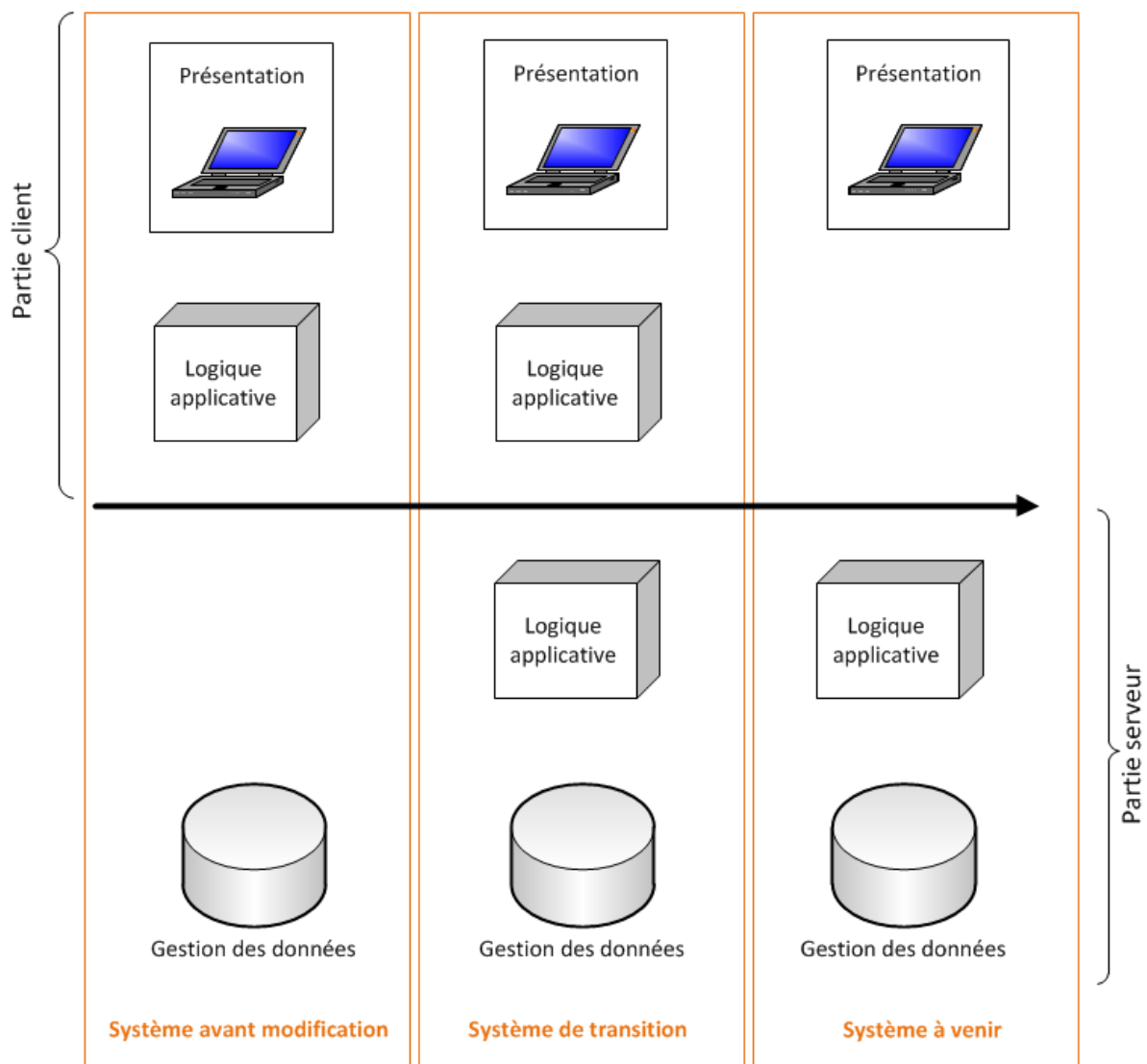


Figure 13 – Évolution de l'architecture logicielle client en client/serveur

La couche présentation

Elle correspond à l'interface homme machine (IHM), c'est-à-dire la partie visible de l'application et celle qui assure l'interactivité avec l'utilisateur. Dans notre cas l'IHM peut être un client riche développé sous Delphi, soit un client léger en PHP pour les solutions existantes et de l'ASP .NET²¹ en client léger pour les nouveaux produits.

La couche des services web

Elle correspond à l'ensemble des fonctions disponibles sur le serveur web. Les clients pourront adressés au serveur les demandes ou requêtes en fonction de ce qui est publié. La couche présentation n'a pas directement accès à la couche métier, elle est contrainte d'utiliser les services web pour communiquer indirectement avec la base de données.

La couche métier

Elle correspond à la part fonctionnelle de l'application, celle qui contient la logique et qui décrit les opérations effectués sur les données.

La couche d'accès aux données

Elle consiste à gérer l'accès aux bases de données et effectuer les opérations demandées par la couche supérieure.

²¹ ASP.NET est un ensemble de technologies de programmation web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites web dynamiques, des applications web ou des web services XML. La technologie est accessible grâce à l'installation d'un serveur web compatible ASP (IIS) ou à l'intérieur de Visual Web Developer Express Edition.

La première difficulté dans la phase de développement a été la mise en conformité des bases de données en fonction de la nouvelle technologie choisie.

Évolution du modèle conceptuel de données

Pour quelles raisons devons-nous spécialement faire évoluer le modèle conceptuel de données (MCD) alors que nous souhaitons réorganiser technologiquement notre architecture logicielle ?

La mise en conformité du modèle de données est nécessaire pour trois raisons :

- il est obsolète et certaines structures ne sont plus compatibles avec la nouvelle technologie,
- il n'a pas été conçu pour gérer de façon dynamique les règles métiers,
- il n'intègre pas certaines notions créées lors de cette évolution, notamment le système de liaison entre un client, une application et ses règles uniques.

J'ai également évoqué précédemment la multitude de bases de données exploitées par les différents applicatifs. La refonte du MCD tel que je l'ai pensée consiste à gérer de façon dynamique et configurable la liste des bases nécessaires pour faire fonctionner un logiciel. Nous réalisons selon les cas des fusions de bases de données lorsque la séparation ne se justifie plus.

La structure relationnelle doit essentiellement évoluer car nous procédons de façon progressive et cohérente à une mise à jour du moteur de base de données MySQL pour chacun de nos serveurs de données. En effet certaines limitations que nous rencontrons avec la suite logicielle existante proviennent de l'ancienneté des versions utilisées.

Lors de l'évolution structurelle des bases de données nous allons intervenir sur la suite logicielle existante car celle-ci exploite toujours directement les données dans le système de transition ([Figure 13](#)).

Préparation d'une plateforme prototype

Avant d'appliquer un certain nombre de modifications ou d'évolutions sur l'infrastructure système actuelle nous avons pris le soin de réaliser une maquette complètement fonctionnelle dans l'environnement qui sera par la suite le modèle pour l'ensemble de la suite logicielle.

Le modèle d'architecture historiquement choisit par le service informatique est le cloud²² par des hébergeurs²³ spécialisés. Nous louons donc ce qu'on appelle des serveurs dédiés²⁴ sur lesquels nous sommes entièrement autonomes. Pour schématiser la principale différence avec l'acquisition de serveur est la localisation et la maintenance physique de celui-ci. Dans un cas le serveur est géographiquement hébergé dans les centres de données²⁵ des hébergeurs dans un autre dans une salle serveur²⁶ dans les locaux de l'entreprise. Le mode de location nous affranchit de toutes les contraintes afférentes à l'hébergement de serveur. En effet, même si nous avons quelques serveurs pour d'autres parties du système d'information, la suite logicielle est entièrement hébergée sur des serveurs dédiés. Ainsi la sécurité des données est assurée.

Nous conservons cette architecture, et pour ce projet nous avons donc sélectionné un serveur dédié sous Windows Serveur, alors que notre flotte actuelle est entièrement sous Linux, afin d'héberger nos applications dotNet.

La préparation d'une plateforme type c'est également un serveur virtuel en interne, sur le serveur ESXI tel qu'évoqué précédemment, et le processus est le suivant :

- un ou plusieurs développeurs demandent à effectuer une nouvelle version du logiciel,
- le noyau (logique applicative) est déployé sur le serveur interne, testé par l'équipe informatique puis par les responsables métiers,

²² Cloud : hébergement de serveur distants par des fournisseurs spécialisés, disponible sur internet avec généralement des bandes passantes disponibles très élevées.

²³ Un hébergeur est une entité qui met en place tout une infrastructure physique complexe et sécurisée pour avoir principalement une activité de location de services.

²⁴ Un serveur dédié, est un serveur physique ou virtuel accessible par l'intermédiaire d'un hébergeur qui le loue.

²⁵ Un centre de données est un site physique sur lequel sont regroupés tous les équipements constituant du système d'information.

²⁶ Une salle serveur est l'équivalent d'une centre de données à échelle beaucoup plus modérée, elle permet de stocker et gérer les serveurs d'une entreprise en respectant plus ou moins de règles de bon conditionnement.

- dans le cas d'une validation le déploiement peut être effectué sur le serveur de production, c'est-à-dire le serveur online hébergé,
- dans le cas d'une recette négative, le développeur est sollicité pour analyser le problème soulevé.

Dans l'architecture conçue nous avons deux types de version à maintenir :

- celles de la partie serveur,
La partie serveur est mutualisée, c'est-à-dire qu'elle nécessite le déploiement sur une seule plateforme.
- celles de la partie cliente.
La partie cliente est soit mutualisée si c'est un serveur web (PHP ou ASP .NET) soit installés sur le poste de chaque utilisateur.

Une fois ces parties préliminaires achevés nous pouvons poursuivre sur la consolidation du produit.

5.2.2 Consolidation du produit

Cette partie correspond au développement sur la continuité du socle de la totalité des fonctionnalités de l'application. Ici nous parlons de consolidation du nouveau produit développé, c'est-à-dire l'application « Association » dont les cahiers des charges fonctionnels et techniques ont été rédigés avant cette consolidation.

Ce produit est entièrement conçu en .NET pour la partie logicielle et exploite une base de données MySQL dont la structure est issu des bases existantes auxquelles nous avons soustrait les éléments spécifiques à la gestion de sinistre puis ajouté de nouveaux éléments propre à la gestion d'une association sportive. Nous avons pris le soin de maintenir un MCD modèle afin d'être dupliqué à chaque nouveau client de ce produit.

Le produit étant entièrement nouveau nous le concevons directement selon le modèle client/serveur en architecture « n-tier » comme évoqué précédemment. Même si techniquement toutes les couches sont hébergées sur le même serveur physique nous nous interdisons d'utiliser directement la couche métier de la couche présentation pour faciliter l'évolution de l'application.

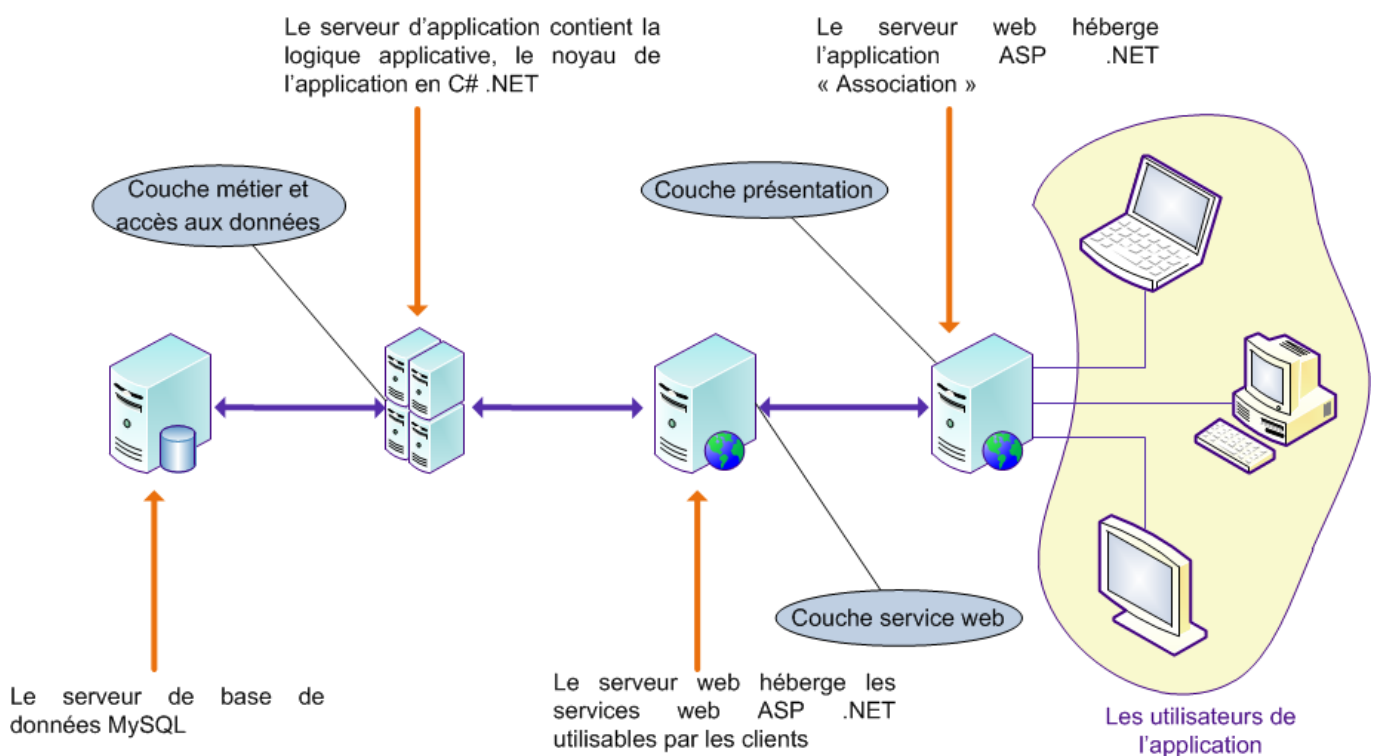


Figure 14 – Présentation de la structure technologique du nouveau produit

La mutualisation de serveur

Dans un objectif de maîtrise des coûts, j'ai effectué des tests de montée en charge, afin de déterminer le nombre de clients qui pourront être hébergés sur le même serveur. En effet, lorsque nous parlons de centralisation des traitements ou d'architecture client/serveur il y a plusieurs points auxquels il faut accorder de l'importance :

- disponibilité de l'application, si le serveur ne fonctionne plus, aucun client ne peut fonctionner (ce qui n'est pas le cas dans une architecture où chaque client contient la logique applicative),
- sécurité de l'application, dans le cas de plusieurs clients hébergés sur le même serveur il faut assurer une sécurité des données et d'accès à celle-ci,
- performance de l'application, il s'agit de maîtriser la montée en charge afin de permettre un accès total par l'utilisateur.

La mutualisation de serveur permet de déterminer, dans un objectif de commercialisation de produit, le point mort de celui-ci. C'est-à-dire le point d'intersection entre le chiffre d'affaire et les coûts nécessaires pour produire ce chiffre d'affaires. Celui-ci aide à fixer le prix du produit, mais aussi à anticiper à quel moment le développement de l'application sera rentabilisé.

La disponibilité et la sécurité de l'application

Lorsque nous mutualisons un serveur physique afin d'héberger plusieurs clients sur ce même support il est nécessaire de maîtriser la disponibilité et la sécurité de l'application.

La location de serveur dédié est le premier élément qui assure un taux de disponibilité avoisinant les cent pourcent. Pour parvenir à ce même taux de disponibilité en internalisant²⁷, l'investissement à effectuer est assez colossal et aucun paramètre n'est à négliger. De l'infrastructure physique, aux compétences de l'équipe, à la liaison internet, tout est de l'importance. C'est pour cette raison que nous sommes restés sur ce modèle d'infrastructure déportée.

Le second élément qui assure la disponibilité de l'application c'est la structure elle-même, où chaque rôle est bien séparé. Il est donc plus accessible d'intervenir sur un des rôles sans affecter les autres.

En dehors de la maintenance applicative conçus par Insiide nous maintenons les systèmes de l'infrastructure déportée et assurons la sauvegarde journalière de l'ensemble des données. Nous avons ainsi une capacité de restauration en cas de problème majeur.

Nous disposons d'ailleurs d'un système automatique mis systématiquement en place lorsqu'on nous avons un nouveau serveur de données. Celui-ci se charge de sauvegarder les données tous les jours et de les restaurer instantanément sur une autre base d'un autre serveur de données. Nous appelons cela une base à « J-1²⁸ ». Celle-ci est utile à plusieurs traitements :

- aux statistiques qui ne sont pas prioritaire en termes de fonctionnement et qui peuvent en cas de problème majeur être arrêté le temps d'un rétablissement d'une plateforme de production,
- aux développeurs qui souhaitent avoir une plateforme identique à celle de production pour certains types de tests, notamment de performances qui peuvent varier entre une utilisation sur le même réseau local²⁹ et une par internet pour exploiter les serveurs dédiés online.

La sécurité est également assurée dès le début de la chaîne, car les services web demandent une authentification pour être utilisés. Dans le cas d'un échec d'identification du client, la demande n'est pas transmise au serveur d'application et celui-ci reçoit une erreur.

Nous étudions d'ailleurs la notion de baie virtuelle³⁰, qui nous permet de sécuriser au maximum les serveurs de base de données, rendu inaccessible par internet. Pour mettre en place cette dernière il faut attendre d'avoir déplacé sur l'ensemble des clients la logique applicative afin qu'ils n'exploitent plus que les services web. C'est un défi technique que nous tendons à atteindre.

²⁷ Internaliser, afin que les serveurs de production appartiennent à l'entreprise et non à des hébergeurs.

²⁸ Les données de la veille sont toujours accessibles sur un serveur différent du serveur de production.

²⁹ Par réseau local j'entends les serveurs de développement qui se trouvent sur le même réseau privé physique que le développeur lui-même.

³⁰ La Baie Virtuelle rassemble virtuellement plusieurs serveurs (quel que soit leur nombre et leur emplacement physique dans les centres de données des hébergeurs) et les connecte à un réseau privé virtuel. Les serveurs peuvent ainsi communiquer de manière privée et sécurisée entre eux.

L'évolution du produit

Le produit « Association », spécialisé dans la gestion d'association sportive de rugby, tend à évoluer sur deux axes :

- l'ajout de fonctionnalité non prévue initialement dans le développement,
- étendre le produit à la gestion d'association d'autres sports, voire même d'autres objectifs associatifs.

Cela dépend entièrement des possibilités commerciales, sur ce point nous avons globalement peu de visibilité. L'objectif pour tous est de finaliser un premier produit sur des sites pilotes qui seront des vitrines pour ces produits et services proposés par Insiide.

En effet un axe transversal consiste à proposer à nos clients et futurs clients un panel de produit que nous pouvons déjà proposer ou qui seront progressivement disponibles. Comme des applications de statistiques, des outils de communication en masse (courriel, sms, etc.), en définitive le contenu du catalogue produit d'Insiide.

L'axe évolutif est transversal à l'ensemble des techniques que nous employons afin de rendre chaque partie comme un module indépendant.

5.2.3 Mise à niveau des logiciels existants

La mise à niveau des logiciels existants, c'est-à-dire précisément l'évolution technologiquement que nous visons tend à concrétiser les éléments suivants :

- mutualisation des fonctions communes,
- déplacer des clients vers les serveurs les traitements lourds en priorité,
- lors du déplacement de fonctionnalité sur le serveur rendre la totalité des règles métier dynamique et configurable par la console d'administration.
- développer une administration unique pour l'ensemble des produits,
- déplacement de l'intégralité de la logique applicative des clients vers les serveurs,

Nous avons pu voir l'architecture en [Figure 14](#) du nouveau produit « Association », celle de la suite logicielle existante est plus complexe car les clients sont plus nombreux et de technologies différentes.

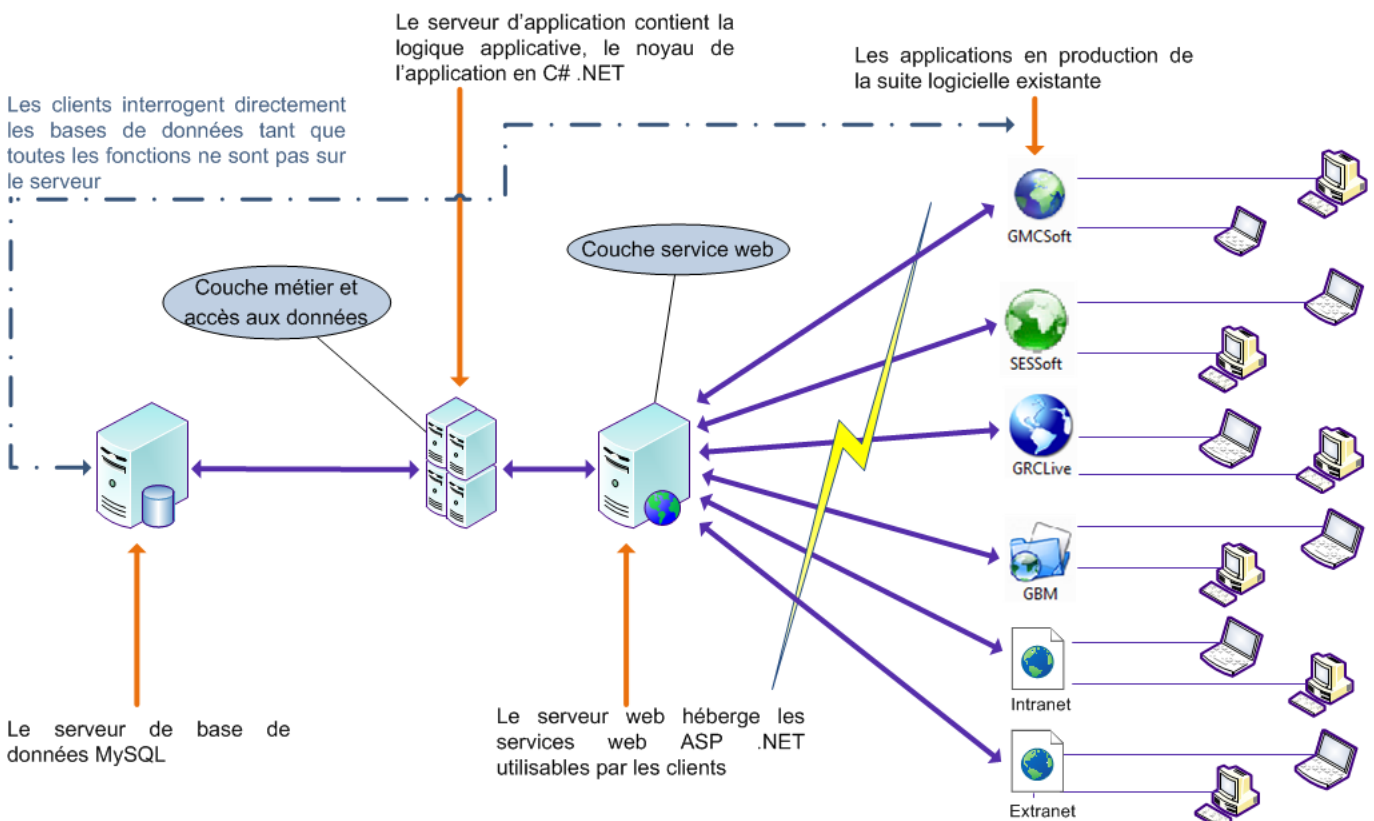


Figure 15 – Présentation de l'évolution de la solution existante

En effet sur la [Figure 15](#) l'infrastructure cliente est beaucoup plus diversifiée, elle est composée de l'ensemble de la suite logicielle existante présentée en [Figure 4](#).

Mutualisation des fonctions communes

Cette partie est transversale à l'ensemble de la suite logicielle existante et répond notamment aux objectifs de diminution des temps de développement et de maintenance des fonctionnalités identiques entre deux technologies.

Nous voulons dans un premier temps maintenir une seule base de gestions des utilisateurs, de leurs droits et rôles. Cependant, nous ne faisons pas de portabilité³¹ de l'existant nous avons décidé de repenser complètement la fonctionnalité.

En effet, ce projet consiste à donner les moyens à Insiide de commercialiser des produits. Hors, par rapport à notre contexte de fonction commune, ce n'est pas simplement des utilisateurs qui utilisent un logiciel, ce sont des clients qui ont à disposition plusieurs modules qui forment une application. Cette application est accessible par des utilisateurs du système à qui nous avons donné les droits correspondants.

Cette notion est centrale dans la mise à niveau technologique en dotNet et nous y avons prêté beaucoup d'attention. Il faut comprendre par cette volonté, qu'aujourd'hui le nombre de bases utilisateurs que nous maintenons est tellement importante que nous aurions du mal à décompter exactement le nombre de système d'authentification différent. Face à cette multiplicité, tous les systèmes qui gravitent autour d'une authentification de l'utilisateur vont être progressivement concernés.

Exemple 1 : les logiciels « GMCSoft » et « l'extranet GMC de suivi de dossier » n'ont pas de gestion commune des utilisateurs.

Comme nous l'avons vu dans les [Figure 4](#) et [Figure 9](#) chaque ensemble de logiciel est composé de trois briques :

- client riche sous Windows, application de gestion collaborative,
- client léger en web, extranet ou intranet de gestion de dossier,
- client léger en web, analyses et statistiques.

Nous parlons d'unité de fonctionnalité entre plusieurs applicatifs de la même activité. Mais ici notre intervention concerna la suite logicielle dans son ensemble et de ce fait tend à mutualiser les similitudes de deux activités du groupe.

Exemple 2 : des utilisateurs du logiciel SESSoft utilisent aussi GMCSoft (essentiellement des responsables ou la direction). Pourtant ceux-ci existent dans deux bases distinctes et sont donc dupliqués.

Ainsi, les différents axes de cette évolution sont :

- refonte du modèle conceptuel de données pour être capable de gérer l'aspect client des applications et de mutualiser la gestion des utilisateurs,
- implémentation des fonctions communes sur le serveur,
- utilisation des fonctions serveur par les clients.

³¹ Une portabilité de code source consiste à réécrire un algorithme d'un langage de programmation vers un autre.

Déplacement des clients vers les serveurs des traitements les plus lourds

Une fois déchargé de l'évolution de l'ensemble de la suite logicielle sur des fonctions communes de fond nous nous sommes attelés au déplacement des traitements les plus lourds en priorité.

Nous poursuivons donc le déplacement de l'intelligence applicative du client vers le serveur comme indiqué sur la [Figure 13](#).

Nous avons déterminés deux traitements spécifiques dont le déplacement est prioritaire :

- Dans l'application GRCLive, un import de contact dans la base de données via un document Excel³²,
- Dans l'application GMCSOFT, la génération de document, rapport d'expertise ou document comptable à partir de l'ensemble des informations renseignées dans le dossier.

Nous avons définis ces deux priorités au vue de plusieurs problèmes qui leur sont liés.

L'import de contact dans le logiciel GRCLive est, dans la suite logicielle existante, directement effectué par le logiciel « client lourd ». Les principaux problèmes retenus sont :

- indisponibilité de l'application lors de l'import, (voire même du poste utilisateur selon les cas),
- manque de fiabilité dû à la sensibilité des problèmes de communication,
- monopolisation quasiment systématique d'un développeur lorsqu'un utilisateur a échoué lors de son import.

Lors de cette injection dans le système d'information l'application traite environ deux à trois milles lignes de fichier Excel, ce qui peut représenter environ dix mille enregistrements en base de données (contacts, société, produits vendus, état des lieux, etc.³³). C'est pour cela qu'une fonctionnalité si volumineuse est déplacée sur le serveur, en voici les avantages :

- l'utilisateur peut lancer une importation et continuer à travailler sur son application cliente,
- le serveur traite plus rapidement et la fiabilité est maximisée car il contacte la base de données dans de meilleure condition que les clients,
- les développeurs n'ont pas à intervenir, où s'ils doivent le faire, ils peuvent directement interroger le serveur pour constater l'avancement et débloquer une situation.

La génération de document pour GMCSOFT est également un traitement lourd et très attendue. En effet, après avoir récolté une quantité d'information sur un dossier (sinistre) durant le travail collaboratif des utilisateurs, le but est de pouvoir générer automatiquement un rapport d'expertise. Celui-ci retrace l'historique du dossier mais aussi des documents comptables. Le rapport d'expertise est un document de cinquante à deux cent pages qui est entièrement conçue par l'application. Cependant les problèmes suivants sont fréquemment rencontrés :

- interruption lors de la génération de document,
- un document généré par un utilisateur n'est pas disponible pour un autre,
- indisponibilité du poste client pendant la génération,
- sollicitation des développeurs lorsqu'un utilisateur rencontre un problème.

Nous avons donc travaillé sur cet axe afin de proposer une génération de document au sens large, à partir de modèle modifiable par l'utilisateur. Celle-ci est directement traitée et archivée par le serveur.

Effectivement au-delà de la génération de rapport d'expertise ou de document comptable nous avons pris les attentes dans leur ensemble afin d'y apporter une réponse définitive :

- génération de document automatisée à une certaine étape d'un dossier,
- archivage des générations effectuées pour qu'elles soient accessibles aux utilisateurs,
- apporter à l'utilisateur une personnalisation maximale, afin d'éviter la sollicitation des développeurs pour apporter une modification à un document (il suffit de modifier le modèle),
- éviter la liaison entre les applications clientes et des logiciels précis sur le poste de l'utilisateur,
- si nous développons à l'avenir des solutions applicatives mobiles ou à destination de plateforme différente de Windows, la génération de document fonctionnelle sur le serveur ne sera pas à redévelopper pour celles-ci.

³² Excel est un tableur de la suite Microsoft Office.

³³ Chaque ligne d'un fichier Excel comporte plusieurs types d'information qui sont stockés à différents emplacement dans la base de données.

Exemple 1 : nos applications Windows (GMCSOFT, SESSOFT, etc.) sont liées à la suite Microsoft Office, Word pour générer des documents. En déplaçant la fonctionnalité sur le serveur, un utilisateur qui ne dispose pas de cette suite installé sur son poste de travail peut tout de même générer des documents.

Nous avons également beaucoup avancé sur la création d'outil comptable notamment, les factures et relances clients.

En effet, nous sommes parti du constat suivant : lorsqu'un gestionnaire de compagnie d'assurance, est réaffecté ou quitte celle-ci, la facturation expédié à son attention est perdu et jamais honorée.

Celui-ci est accompagné de fait marquant, au moment où nous avons abordé ce sujet, pour la seule activité de GMC Cabinet d'expertise, les factures en attente de règlement sont de l'ordre d'un virgule deux millions d'euros. Cela nous confirme donc que c'est une situation largement répandu et qui a des impacts significatifs sur la trésorerie de l'entreprise.

L'objectif est de concevoir un système de relance au maximum automatisé afin de faire un dossier comprenant une lettre récapitulative, les factures concernées et le récapitulatif de chaque dossier. Ainsi le service comptable du groupe n'a plus à se soucier d'effectuer ce travail de fond absolument gigantesque, il doit simplement nous transmettre les données variables qu'il veut traiter (fréquence des relances, nombre de mois de retard de paiement) et pouvoir obtenir un historique des traitements.

Nous comprenons donc ici que ces fonctionnalités sont prioritairement traitées pour des raisons techniques fortement appuyés par des raisons décisionnelles et budgétaires. Je dirai que toute la complexité de ce projet consiste à parvenir à allier différentes causes pour servir la ligne directrice générale.

Gestion dynamique des règles métier

Nous appelons gestion dynamique des règles métier, lorsque le code source de l'application est dénué de toute spécificité liée à l'activité d'une entreprise. Nous avons malheureusement trop rencontré, pour des raisons de facilité et rapidité de mise en place, des algorithmes inutilisables pour d'autres logiciels ou d'autres clients à cause de règles qui ne s'appliquent pas à tous.

L'objectif est que l'ensemble des paramètres propres à un client soit entièrement stocké en base de données afin de n'introduire dans le code source aucune spécificité.

De plus étant donné que nous déplaçons l'intelligence applicative du client vers le serveur ([Figure 13](#)), des fonctionnalités qui étaient avant complètement distincte seront donc unifiés sur le serveur. Hors le serveur doit être capable de gérer plusieurs cas de figure.

Nous utilisons pour cela la notion d'héritage³⁴ lors de la programmation de la partie serveur. Ce principe nous permet de créer des objets génériques et des objets spécifiques qui en découlent.

Exemple 2 : nous pouvons créer l'arborescence suivante :

« Dossier » qui va contenir l'ensemble des caractéristiques générales,

« Dossier Grand Risque » qui va contenir les spécificités pour GMC,

« Dossier Bien de masse » qui va contenir les spécificités pour SES.

Ainsi de suite pour toutes les activités pour lesquelles nous avons un « dossier » ou une « affaire » à traiter avec des caractéristiques spécifiques.

Dans cette architecture cela veut dire que le client doit passer au serveur lors de sa requête l'ensemble des données variables qui sont attendues en retour.

³⁴ Principe qui permet de créer un objet qui dérive d'un autre. Par exemple un objet « véhicule » contient un ensemble de propriétés et de fonctions. L'objet « voiture » qui dérive de « véhicule » contient les propriétés de ce dernier plus des propriétés qui lui sont propre. Puis un objet « camion » qui dérive également de « véhicule » aura d'autres propriétés différentes de la « voiture ».

Développer une administration unique pour l'ensemble des produits

La mutualisation des fonctions communes (quel que soit la technologie cliente utilisée) et la gestion dynamique des règles métier nous permet de développer une console d'administration globale à l'ensemble des produits proposés par Insiide.

L'administration centralisée peut être scindée en deux niveaux :

- un premier niveau technique,
- un second niveau destiné aux responsables métier.

L'administration technique

Celle-ci tend à gérer les aspects techniques de la suite logicielle, ce que l'on appelle plus communément une « console administrateur ». Par administrateur nous entendons les membres de l'équipe informatique capable de gérer le système d'information.

Les fonctionnalités sont hiérarchisées de la façon suivante :

- les applications disponibles,
- les clients d'Insiide,
- les accès aux bases de données métiers,
- les utilisateurs,
- les droits d'accès gérés dans les applications,
- le menu des applications,
- l'historisation des événements systèmes.

Cette base est le point d'entrée de l'ensemble de nos applications en mode client/serveur. Elle contient notamment l'ensemble des accès aux bases de données métiers, ce qui en soit peut représenter une faille de sécurité malgré le fait que nous ayons pris un maximum de précaution. Certaines informations sont cryptés et pas seulement les mots de passes. Les serveurs hôtes³⁵, les noms d'utilisateurs et les mots de passes font parties des informations cryptés.

Cependant des sous parties qui ne nécessite pas d'être administrateur pour être maintenu seront, comme prévue, délégués aux responsables métiers.

L'administration métier

L'administration métier consiste à gérer une partie des utilisateurs.

Exemple 1 : modifier les droits d'une personne afin qu'elle puisse avoir un accès en écriture à un élément alors qu'elle ne l'avait qu'en lecture.

Nous déléguons progressivement ces fonctionnalités, et nous nous assurons que les responsables métiers ont un minimum de formation pour comprendre les impacts de leur modification.

Cela a clairement pour but de décharger Insiide de certaines tâches mais il faut faire attention de le réaliser dans de bonne condition afin que cela ne crée pas par la suite plus de travail à cause de problème non prévus. En effet, étant donné que la console d'administration est un développement récent, elle n'intègre sûrement pas tous les cas de figure de manipulation possible par un utilisateur.

Au-delà de gestion système, comme la gestion des utilisateurs, les responsables métiers pourront modifier des éléments propres au métier de la structure dont ils sont responsables. Les propriétés modifiables sont celles qui ont été portées en dotNet, et dont les règles métier sont dynamiques.

Exemple 2 : pour le système de gestion des relances de GMC, le responsable métier peut paramétrer la périodicité des relances, la date des factures prises en compte (les factures expédiées depuis plus de six mois) et bien d'autres.

Pour chaque fonctionnalité nous avons des paramètres dynamiques qui seront modifiable. Comme précisé afin de pouvoir administrer les logiciels sans l'aide de développeur il faut en arriver au déplacement de la logique applicative vers les serveurs. Mais le projet que nous réalisons n'intègre pas entièrement cette tâche qui est expliqué ci-après.

³⁵ Un serveur hôte est un ordinateur puissant qui délivre des services, comme par exemple serveur de base de données, serveur de fichiers, etc.

Déplacement de l'intégralité de la logique applicative des clients vers les serveurs

L'objectif final est de parvenir à une suite logicielle client/serveur, avec d'une part les serveurs, capables de gérer plusieurs cas de figure et d'autre part les clients applicatifs client léger ou client riche qui exploitent les possibilités des serveurs.

Mais l'ampleur de cette tâche est tentaculaire au vue de l'existant créé en plusieurs années. Certaines fonctionnalités sont d'ailleurs très peu maîtrisées des développeurs car elles ont pu être créés par des collaborateurs qui ne font plus partie du groupe et qui sont donc restés statiques et utilisés telles quelles.

De plus les applicatifs clients, autrement dit les interfaces utilisateurs (IHM) sont eux spécifiques au métier qu'ils servent.

En effet, nous avons analysé l'expérience suivante : lorsque nous utilisons une interface (complexe) pour plusieurs besoins spécifiques, la durée de maintenance est finalement plus importante que si nous avons réalisés deux interfaces distinctes. Ceci s'explique par des comportements souhaité complètement différent entre deux métiers. Donc, le temps gagné lors de la création de l'interface qui consiste à utiliser une interface existante est perdu au fil du temps.

Nous déplaçons effectivement la logique applicative, les traitements qui suivent les actions utilisateurs, mais les interfaces restent spécifiques. Nous tendons même à scinder quelques interfaces en deux pour les raisons évoqués précédemment.

De plus lorsqu'un développeur doit retoucher une interface qui implique plusieurs métiers il est nécessaire de maîtriser plusieurs sujets différents. Dans une équipe relativement petite, polyvalente et surtout conceptrice des applications cela ne pose pas trop de problème. Par contre dans un objectif d'anticipation d'une augmentation d'effectif cela pose de réels problèmes lié à la connaissance de plusieurs activités distinctes.

Nous avons donc tout intérêt à planifier :

- le déplacement de l'intégralité de la logique applicative vers les serveurs,
- le nettoyage des interfaces clientes lorsque cela est nécessaire.

5.3 Recette et mise en production

Avant de réaliser un bilan du projet nous avons mené des recettes de nos différentes productions.

Les recettes ont été scindées en plusieurs parties logiques, fonctionnelles et techniques, en collaboration avec les responsables métier avec qui nous avons rédigés les cahiers des charges fonctionnels notamment.

Comme nous pouvons le voir dans la planification, [Figure 10](#), nous avons accordé un mois à la recette et à la mise en production finale. Pour un projet d'un peu plus d'un an, c'est un délai un peu court mais nous avons la particularité de pouvoir découper en plusieurs responsables fonctionnels la recette.

Avant toute chose et le lancement de la recette fonctionnelle, nous avons entrepris notre propre série de préparatifs préliminaires afin de garantir que le livrable est conforme à la demande. Nous nous assurons que la recette peut être opérée dans de bonnes conditions.

J'ai donc organisé une réunion préparatoire avec les différents responsables afin de valider leurs disponibilités et la méthode de validation en trois temps :

- le responsable fait un recueil d'anomalies et me les communique,
- les corrections sont apportées et validées par le responsable qui peut continuer son recueil pendant ce temps,
- le responsable a terminé le recueil d'anomalie, les corrections sont terminées, il valide définitivement le livrable.

Un cahier de recette est transmis afin d'éviter que les responsables effectuent des remontées qui ne sont pas des anomalies. Le but est d'éviter que des demandes d'évolution soient sans cesse formulées.

Nous n'avons pas de notion de pénalité (de retard notamment) puisque nous faisons une recette avec des interlocuteurs internes ou intervenants externes en partenariat, par contre nous avons tout de même fixé un planning cohérent pour parvenir à la validation.

La recette fonctionnelle

La recette fonctionnelle a pour but la validation des fonctionnalités exprimées dans le cahier des charges et détaillées dans les spécifications fonctionnelles. La validation se concentre sur trois aspects du livrable :

- les tests fonctionnels, à partir du cahier de recette, vérifient que les différents modules correspondent aux exigences,
- les tests d'un point de vue des cas d'utilisation, vérifient que les grands principes sont valides par rapport aux actions qui seront effectués en production,
- les tests de performance et de robustesse, vérifient que les modules n'ont pas des problèmes de stabilité, fiabilité dans le temps et au vue d'une augmentation de la quantité de données à traiter.

Le protocole de recette est garanti par la remontée d'anomalies par l'intermédiaire du gestionnaire de ticket de l'outil de gestion de projet. Celui-ci garantit le suivi et la qualité de la recette.

Nous avons mis en production, respectivement, les évolutions pour la suite logicielle existante et pour le nouveau produit lorsque nous avons la validation des responsables métier et de la direction en dernier lieu.

La recette technique

La recette technique a pour but la validation des exigences par rapport à l'architecture mise en place. Elle doit garantir la performance de l'application au vue des besoins énoncés par les responsables métier.

J'ai participé à cette recette et le directeur informatique en a validé définitivement le contenu sur les axes suivant :

- réponse aux attentes fonctionnelles,
- disponibilité et maintenance,
- sécurité et traçabilité,
- performance et évolution.

La liste des tests, que comporte le cahier de recette, est inévitablement limitée : certaines anomalies apparaîtront lors de la mise en production du site pilote. Il est donc préférable d'anticiper les problèmes éventuels en termes de ressources humaines disponibles lors de la mise en production des différentes solutions. En effet, nous ne pouvons a priori pas envisager la difficulté des corrections à mettre en œuvre, il faut donc s'assurer que nous resterons dans des délais acceptables pendant la phase de recette finale.

6. Conclusion

6.1 Bilan du projet

L'ensemble du projet porte sur la faculté d'un service, transformé en entreprise indépendante, à se restructurer et à évoluer pour servir non seulement les intérêts internes des entreprises qui l'ont fait naître, mais aussi de rechercher des nouvelles sources de revenus externes.

Insiide, et a fortiori le groupe, a différentes sources de compétences à l'origine de son savoir-faire. L'équipe informatique maîtrise tout d'abord l'édition de logiciel à proprement dit, puis sa mise en œuvre, son intégration et sa personnalisation. Cela veut concrètement dire que nous sommes entièrement autonomes dans la livraison d'application, standard ou personnalisée à de nouveaux clients.

Les différents acteurs qui ont participé à la création du système d'information, les dirigeants, les responsables et l'équipe informatique, ont défini les objectifs et les contraintes de ce projet afin d'obtenir chronologiquement un autofinancement d'Insiide et un retour sur investissement progressif des développements internes.

Valorisation commerciale

Le potentiel du nouveau marché est important, nous avons un prévisionnel sur des années complètes de trente à soixante nouveaux clients. À moyen ou long terme nous pourrions concrétiser un partenariat afin d'équiper systématiquement les associations sportives. Des négociations ont été entreprises avec des organisations afin de formaliser et unifier la communication entre les clubs et les fédérations. Différentes déclinaisons de ce produit pourront être réalisées afin d'augmenter le nombre de clients concernés, notamment en nous spécialisant dans d'autres sports.

Création d'une culture de gestion de projet professionnelle et non plus gérée au fil de l'eau

Par ailleurs, nous avons une source de compétence et d'expérience que nous ne retrouvons pas communément dans une société de services et d'ingénierie informatique. Grâce au groupe spécialisé dans l'expertise (activité de GMC), nous avons un retour d'expérience significatif sur des projets informatiques (sur des sujets variés) qui se retrouvent en situation d'échec gérés comme des sinistres. Nous avons donc plusieurs illustrations de conduite à éviter, d'erreur commise lors de la gestion de projet qui mène parfois à la dissolution de la société gestionnaire dans les cas les plus graves.

Gain de performance

L'évolution de la suite logicielle existante a également été bénéfique pour le groupe, notamment grâce à la fonctionnalité de relance de facturation client, qui a permis de recouvrer plus de dix pourcent de chiffre d'affaire en attente de règlement en seulement un mois d'utilisation.

Nous avons également sensiblement réduit la quantité de bogue³⁶ décelé avec seulement quelques fonctions communes redéveloppées dans le nouveau langage. La performance du portage est mesurée efficacement par cet indicateur. La qualité des développements, s'explique notamment par l'expérience acquise et l'amélioration des codes existants.

Insiide commence à ressentir les bénéfices de l'évolution en termes de charge de travail. Les collaborateurs ont plus de temps pour poursuivre les modifications de fond. Ce processus progressif est gigantesque à mener, et je trouve difficile d'en estimer sa réelle fin. Disons que nous avons réalisé sur la suite logicielle existante dix pourcent du travail en un an de projet.

Bien que nous avons mis en place des prérequis structurel et créé de nouveau produit l'évolution de la totalité de la suite logicielle sur le modèle d'architecture définitif va être réalisé en plusieurs années.

³⁶ Bogue (de l'anglais bug) : défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement.

Gestion des ressources

Grâce à ce projet les développeurs ont développés et diversifiés leurs compétences en maîtrisant un autre langage de programmation. Cela dynamise et motive l'équipe qui a désormais la possibilité de varier son temps de travail en réalisant des modifications intéressantes plutôt que de mener des corrections répétitives avec les mêmes techniques et les mêmes problèmes rencontrés.

De ce savoir-faire et de cette expérience nous avons su répartir notre charge de travail. Nous l'avons réalisé de manière à ce que l'activité principale (GMC/SES), ne connaisse pas de baisse de production tout en veillant à faire progresser Insiide. C'est-à-dire apporter des évolutions de fond, structurelles et techniques, tout en créant un nouveau produit.

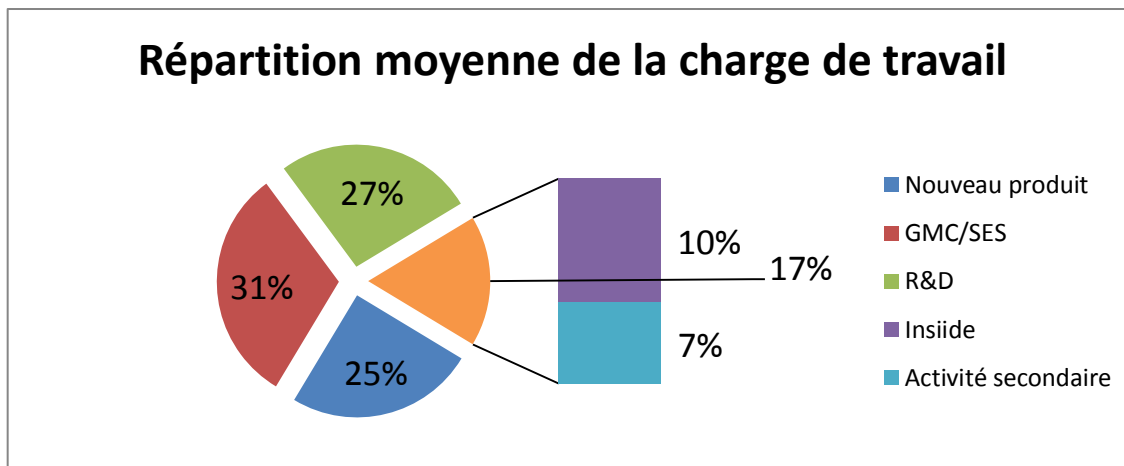


Figure 16 – Répartition moyenne des temps passés

Les éléments essentiels mis en évidence dans ce graphique montrent que nous avons trois principaux axes pendant le projet :

- l'évolution de la suite logicielle pour GMC/SES,
- la recherche et développement (R&D),
- la création de nouveau produit.

Tout ceci explique comment nous avons été capables de nous diversifier dans plusieurs secteurs d'activités tout en produisant un résultat satisfaisant.

Produit fini

Le nouveau produit « logiciel de gestion d'association sportive de rugby », a été commercialisé en quatre exemplaires, en plus des contrats de maintenance souscrits par ces derniers clients.

Le résultat obtenu est principalement dû au fait que chacun des acteurs a une idée précise de ses propres besoins. La valorisation du travail accompli peut maintenant se poursuivre vers une réelle commercialisation encouragée en cela, par les expériences positives avec deux clients intéressés par le produit.

Concernant l'atteinte de tous les objectifs stratégiques certains sont difficilement évaluables par manque de recul nécessaire pour évaluer le degré de réussite de ces derniers. Cela concerne essentiellement la rentabilisation d'Insiide qui n'est pas atteinte au moment où nous faisons ce bilan. Le gain apporté à la structure n'en reste pas moins positif du fait de la réduction de complexité des systèmes, des retombées économiques des structures gérées et des ambitions au long terme d'Insiide.

6.2 Les apports personnels

Lors de ce projet j'ai pris du recul sur ce qu'implique la conception d'application afin de faire évoluer une entreprise qui a pour intention de mettre sur le marché des solutions jusqu'ici développées et utilisées en interne.

Nous pouvons constater un équilibre entre la partie technique et les parties stratégiques puis organisationnelles. Ainsi, la complexité du projet se situe dans la polyvalence à mettre en œuvre afin d'apporter une solution conjointe à ces deux sujets.

J'ai progressivement pris conscience des responsabilités qui m'ont été confiées. Mener ce projet m'a permis de prendre confiance en moi tout en appliquant avec rigueur une ligne directrice. Celle-ci s'étend de l'organisation d'une équipe à une communication avec différents interlocuteurs. Il est nécessaire, dans les situations que j'ai rencontrés, de faire preuve de discernement afin de prioriser à juste titre les objectifs et la façon de les atteindre. J'ai souhaité prendre en compte les problèmes rencontrés dans leur intégralité afin de définir des bases solides aussi bien sur le plan technique qu'organisationnel. Mon sens de l'organisation et mes compétences techniques m'ont amené à prendre du recul afin d'apporter les solutions adéquates. Celles-ci sont d'ailleurs en constante évolution car je pense qu'il est indispensable, dans notre profession, d'être à l'écoute des tendances afin de pouvoir les mettre au service des besoins de l'entreprise. Mes acquis m'ont assuré la légitimité nécessaire pour mener à bien ce projet, non seulement auprès des différents interlocuteurs (demandeurs), mais aussi avec les collaborateurs du service informatique.

Cette expérience réalisée chez Insiide m'a permis d'appréhender une gestion de projet d'envergure. L'édition de logiciel est un métier que j'exerce depuis plusieurs années et il m'a paru nécessaire de m'orienter vers une autre activité novatrice. Le but est de me diversifier tout en ayant la possibilité d'exploiter les compétences développées depuis mon début de carrière.

À l'issue de ce projet j'ai donc annoncé mon départ d'Insiide pour servir ces objectifs plus personnels.

Avant de partir, j'ai bien évidemment pris les précautions nécessaires pour que les collaborateurs aient toutes les capacités et les informations pour poursuivre l'évolution et faire progresser la structure. D'ailleurs, une gestion de projet et une communication comme celle mise en place limite fortement les perturbations qu'il peut y avoir lorsqu'un collaborateur quitte la structure.

J'ai eu l'opportunité d'intégrer en tant que « Responsable informatique et Responsable R&D » une entreprise, IBITEK, dans l'industrie lourde. Un secteur d'activité, dont le cœur de métier m'est inconnu, mais dans lequel je mets en œuvre mes compétences et connaissances notamment développées par le projet présenté. Le potentiel de cette entreprise et la possibilité de réaliser des projets d'envergures m'ont particulièrement intéressé et motivé pour intégrer ce poste.

C'est tout de même une décision difficile que de quitter Insiide après une telle confiance, qui a été mutuelle, et un travail conséquent que j'ai tenté de vous présenter dans ce mémoire.

7. Glossaire

Page	Mot ou expression	Définition
10	¹ Suite logicielle	Une suite logicielle est un ensemble de programmes informatiques, offrant des fonctionnalités liées, et possédant souvent une interface graphique plus ou moins similaire.
11	² Microsoft dotNet	Microsoft .NET est un ensemble de produits et de technologies édité par Microsoft. Il permet notamment de développer en plusieurs langages de programmation.
11	³ Delphi	Delphi est à la fois un environnement de développement intégré et un langage de programmation orienté objet. Édité par Borland il implémente la version orientée objet du Pascal.
13	⁴ Client léger	Il existe plusieurs sortes de client léger mais ici nous parlons de client léger pour un logiciel qui s'exécute dans un navigateur web.
13	⁵ Client lourd	Un client lourd est un logiciel qui comprend l'ensemble des fonctionnalités et qui n'a besoin d'un serveur que pour stocker les données. Il embarque donc les interfaces graphiques et l'intelligence applicative.
15	⁶ Client / Serveur	Une approche client/serveur consiste à scinder les rôles client et serveur en deux applications distinctes. L'une (le client) envoi des requêtes, l'autre (le serveur) attend les requêtes des clients et y répond.
22	⁷ ETL	Extract-Transform-Load (ETL), extracto-chargeur, est un système permettant d'effectuer des synchronisations massives d'une source vers une autre, en général une base de données.
23	⁸ PHP	Hypertext Preprocessor (PHP) est un langage de programmation libre utilisé pour produire des pages web dynamiques. C'est un langage interprété par un serveur web.
36	⁹ Microsoft Action Pack	Pack de licence souscrit auprès de Microsoft par le biais d'un abonnement qui donne droit à l'utilisation à des fins de tests et de développements certains produits.
36	¹⁰ Microsoft Partner Network	Réseau de partenaire de Microsoft qui donne droit à un certain nombre d'avantages, comme être référencé parmi les partenaires officiels. Il existe plusieurs niveaux de partenariat.
36	¹¹ Visual Studio 2010	Visual Studio 2010 est un ensemble complet d'outils de développement pour la plateforme dotNET.
36	¹² Windows Server 2008	Windows Server 2008 est un système d'exploitation édité par Microsoft orienté serveur.
37	¹³ Compilation	La compilation d'une application consiste à transformer le code source écrit dans un langage de programmation en un autre langage informatique qui puisse être exécuté sur une plateforme cible (le poste de l'utilisateur).
38	¹⁴ Hyperviseur	Plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une machine physique en même temps.
38	¹⁵ Machine virtuelle	Une machine virtuelle est à l'intérieur d'un hyperviseur, une reproduction de composant d'un appareil informatique afin de faire fonctionner le système d'exploitation classique.
38	¹⁶ VMware ESXI	C'est un hyperviseur édité par la société VMware, il en existe d'autres.
38	¹⁷ Starteam	Système de gestion de versions de code source d'une application, édité par Borland.
38	¹⁸ SVN	Système de gestion de versions édité par Apache et BSD, édité par CollabNet.
39	¹⁹ Architecture n-tier	Architecture n-tier : architecture qui consiste à séparer en différentes couches les différentes parties qui constitue une application. Cette logique consiste à donner un ordre aux couches

		et où le développeur s'interdit à invoquer une couche qui n'est pas directement inférieure ou supérieure.
39	²⁰ Couche (applicative)	Couche (applicative) : chaque couche communiquent entre elles pour proposer un ensemble de service.
40	²¹ ASP .NET	ASP.NET est un ensemble de technologies de programmation web créé par Microsoft. Les programmeurs peuvent utiliser ASP.NET pour créer des sites web dynamiques, des applications web ou des web services XML. La technologie est accessible grâce à l'installation d'un serveur web compatible ASP (IIS) ou à l'intérieur de Visual Web Developer Express Edition.
41	²² Cloud	Cloud : hébergement de serveur distants par des fournisseurs spécialisés, disponible sur internet avec généralement des bandes passantes disponibles très élevées.
41	²³ Hébergeur	Un hébergeur est une entité qui met en place tout une infrastructure physique complexe et sécurisée pour avoir principalement une activité de location de services.
41	²⁴ Serveur dédié	Un serveur dédié, est un serveur physique ou virtuel accessible par l'intermédiaire d'un hébergeur qui le loue.
41	²⁵ Centre de données	Un centre de données est un site physique sur lequel sont regroupés tous les équipements constituant du système d'information.
41	²⁶ Salle serveur	Une salle serveur est l'équivalent d'une centre de données à échelle beaucoup plus modérée, elle permet de stocker et gérer les serveurs d'une entreprise en respectant plus ou moins de règles de bon conditionnement.
43	²⁷ Internaliser	Internaliser, afin que les serveurs de production appartiennent à l'entreprise et non à des hébergeurs.
43	²⁸ J-1	Les données de la veille sont toujours accessibles sur un serveur différent du serveur de production.
43	²⁹ Réseau local	Par réseau local j'entends les serveurs de développement qui se trouvent sur le même réseau privée physique que le développeur lui-même.
43	³⁰ Baie virtuelle	La Baie Virtuelle rassemble virtuellement plusieurs serveurs (quel que soit leur nombre et leur emplacement physique dans les centres de données des hébergeurs) et les connecte à un réseau privé virtuel. Les serveurs peuvent ainsi communiquer de manière privée et sécurisée entre eux.
45	³¹ Portabilité	Une portabilité de code source consiste à réécrire un algorithme d'un langage de programmation vers un autre.
46	³² Excel	Excel est un tableur de la suite Microsoft Office.
46	³³ Enregistrements	Chaque ligne d'un fichier Excel comporte plusieurs types d'information qui sont stockés à différents emplacement dans la base de données.
47	³⁴ Héritage	Principe qui permet de créer un objet qui dérive d'un autre. Par exemple un objet « véhicule » contient un ensemble de propriétés et de fonctions. L'objet « voiture » qui dérive de « véhicule » contient les propriétés de ce dernier plus des propriétés qui lui sont propre. Puis un objet « camion » qui dérive également de « véhicule » aura d'autres propriétés différentes de la « voiture ».
48	³⁵ Serveur hôte	Un serveur hôte est un ordinateur puissant qui délivre des services, comme par exemple serveur de base de données, serveur de fichiers, etc.
51	³⁶ Bogue	Bogue (de l'anglais bug) : défaut de conception d'un programme informatique à l'origine d'un dysfonctionnement.

8. Table des illustrations

Figure 1 – Représentation des principales activités et de leurs répartitions.....	6
Figure 2 – Répartition du chiffre d'affaire du groupe.....	7
Figure 3 – Métier de Gestion de sinistre.....	8
Figure 4 – Suite logicielle existante.....	10
Figure 5 – Évolution de l'organigramme d'Insiide.....	11
Figure 6 - Les deux composantes du projet.....	12
Figure 7 – Organisation d'Insiide sans gestion de projet.....	16
Figure 8 – Organisation d'Insiide avec gestion de projet.....	19
Figure 9 – Architecture des technologies existantes.....	24
Figure 10 – Planification prévisionnelle du projet.....	26
Figure 11 – Architecture schématique d'une solution client/serveur.....	33
Figure 12 – Évolution du déploiement d'application.....	37
Figure 13 – Évolution de l'architecture logicielle client en client/serveur.....	40
Figure 14 – Présentation de la structure technologique du nouveau produit.....	42
Figure 15 – Présentation de l'évolution de la solution existante.....	44
Figure 16 – Répartition moyenne des temps passés.....	52

9. Annexe

I. Couverture fonctionnelle

Comme précisé dans l'objectif d'unification des développements, [1.4.1 Du point de vue du groupe page 14](#), nous avons réalisé un document récapitulatif de la couverture fonctionnelle.

Nous avons mis en évidence deux parties :

- définition du périmètre fonctionnel par module et par fonction,
- couverture fonctionnelle par application.

Définition du périmètre fonctionnel par module et par fonction

Celle-ci précise les modules que nous pouvons dégager de la suite logicielle existante ou des produits à venir. Chaque élément est composé de fonctions est de leurs descriptions.

Cela nous a donc permis de faire l'inventaire des modules couvert tout en les décrivant afin de déterminer leur caractère spécifique à un métier ou générique (c'est-à-dire applicable à d'autres activités).

Couverture fonctionnelle par application

Celle-ci nous permet de définir le niveau d'avancement d'une fonctionnalité par logiciel.

Nous avons trois types de logiciels :

- ceux à développer court terme,
- ceux à développer long terme,
- ceux de la suite logicielle existante.

Nous avons donc qualifié la priorité de chaque fonction ainsi que sa nécessité pour une application.

De cette approche fonctionnelle nous avons pu prioriser et définir le planning prévisionnel du développement.

Lexique	
	Fonctionnalité existante
	Fonctionnalité à valider
	Fonctionnalité manquante
	Pas de besoin







































































































































DEFINITION DU PERIMETRE FONCTIONNEL PAR MODULE ET PAR FONCTION

Lexique priorité	
	Priorité 1
	Priorité 2
	Priorité 3

COUVERTURE FONCTIONNELLE PAR APPLICATION INSIDE

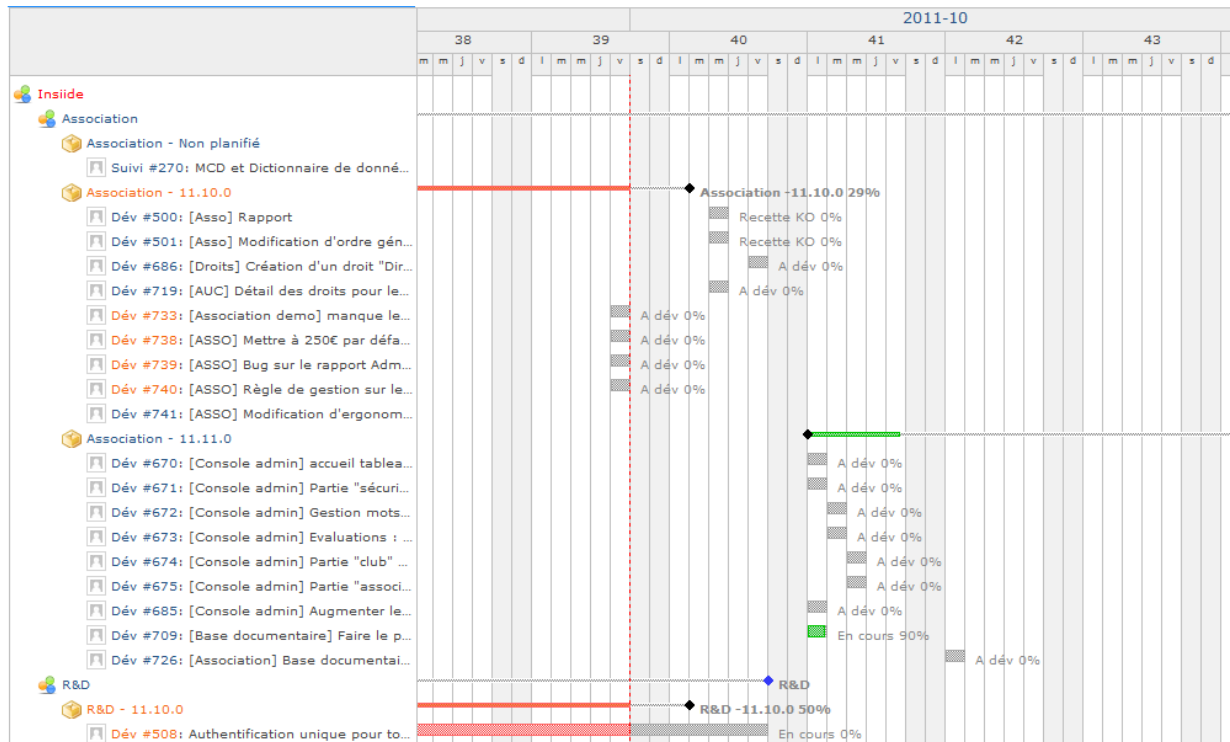
MODULE	MICRO FONCTIONS	DOMAINE FONCTIONNEL	DESCRIPTION DE LA FONCTION	DONNEES CIBLE LIEES AU MODULE	PRIORITE	Nouveau produit à court terme		Nouveau produit à long terme			Logiciels existants					
						Juridique	Association	GEM	Commerce	Marketing	GMCSft	SESSft	GEM	GRCLive	Open Avocat	
 Gestion des droits	Gestion des utilisateurs	Gestion des droits d'accès (données et fonctions)	Attribution de droits sur l'application et sur les données (restriction d'accès, lecture, modification, suppression, ajout)	Toutes les fonctionnalités, toutes les données, tous les reportings												
	Multisite	Gestion des utilisateurs sur plusieurs sites	L'accès à l'application est lié à la notion de sites multiples au sein de la même société.	Toutes les fonctionnalités, toutes les données, tous les reportings												
 Gestion des dossiers	Dossiers	Qualification d'un dossier (adapté en fonction du métier)	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Pour le métier de la Gestion de projets, le domaine comprend la fonction de gestion de portefeuille : Vue synthétique et analytique d'un regroupement de projets.	Les attributs des dossiers varient en fonction de la nature du métier : affaires commerciales, projet, affaire juridique, marketing. On y retrouve l'ensemble des données déclaratives qualificatives (y compris la saisie d'objectifs, de priorisation, de risque, ... quand ces attributs ne sont pas calculés).												
	Nommage, Référentiel et codification	Procédure d'attribution automatique d'une nomenclature selon règle métier (personnalisable)	Génération automatique de codes ou de nom.	Référence dossier, référence document, code campagne marketing, code retour marketing, code comptable, etc.												
 Gestion des contacts	Contacts associés à des rôles	Gestion des contacts (sociétés ou individus) par rôle (interne, externe, clients, frs, Ressources, Oble marketing, dossier (propriétaire, valideur, Suivi, ...))	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Interaction avec les dossiers (rôles), les sociétés, le requêtage	Le MCD peut faire contenir dans la même table l'ensemble des individus impliqués dans le dossier : utilisateurs, contacts (liés aux sociétés ou non), ressources humaines (interne ou externe)												
	Gestion des doublons	Alerte sur création de doublons (items qualifiant à définir)	Génération d'un message d'alerte à minima à la création d'un enregistrement en doublon, blocage de la saisie (en fonction des choix client)	Lien de la société à la création de la base de données. Société : attribut idéal pour la gestion des doublons : N° SFRET. Individus : nom, prénom, rôle(?)												
 Gestion des activités	Conduite des activités par gestion de tâches internes et actions externes	Activités chapeau à réaliser (tâches internes pour la production et le suivi d'un dossier; et actions externes pour les activités liées à des tiers) Une activité est rattachée à un dossier, à un dossier.	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Création d'une interaction avec le dossier (affectation d'activités), l'agenda (marquage chronologique), avec les ressources (disponibilité), avec les objets inclus dans le workflow si automatisé (via le statut) - facturation par exemple.	Concernera tous les dossiers quelle que soit leur nature métier. Exemple d'attributs : description, code, catégorie, statut, date de début et de fin; ou d'objets liés : ressources humaines, documents, contacts,												
	Gestion des Workflow	Automatisation des successions d'étapes prédéfinies par le métier	Les activités peuvent être reliées entre elles par des règles métier. Le workflow y rajoute des séquences, peut être automatisé et peut indexer les activités chronologiquement.													
 Gestion du temps	Agenda	Marquage des activités sur support chronologique (statique)	Peut être un simple calendrier personnel (utilisateur) ou un planning partagé (fonction des droits) Droit unique utilisateur : Visualisation des activités (à rattacher à la gestion des activités) dans un calendrier													
	Planification	La planification reprend les notions de "simulation" : simulation sur la disponibilité des équipes, sur l'avancée du planning (dépassement), sur le respect des délais de livraison.	Analyse de l'impact de nouveaux projets sur des données réelles: budgets, charges, délais. Simulation avec calcul d'indicateurs et marquage dans le calendrier													
	Temps	Evaluation du temps passé par les collaborateurs sur un dossier (activité)	Saisie des feuilles de temps. Rattachement aux utilisateurs, dossiers, activités (potentiel), facturation.													
	Chronomètre	Mesure en temps réel du temps passé sur une tâche	Utilisation d'un chronomètre (le "stop" génère une ligne dans la feuille de temps)													
 Gestion des revenus	Gestion des devises	Conversion automatique des devises (taux de change journalier)	Lien avec fournisseur													
	Avant-Vente : Devis	Lié à un identifiant mais non comptable. Peut être lié à des charges (gestion de projet par ex.). Nommé "provision" aujourd'hui	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Interaction avec les dossiers (historisation, qualification)	Liées au référentiel produit et aux contacts Ne peut concerner que les dossiers de nature commerciales (Ex : Avocat = Proposition de services)												
	ADV : Commandes	Génération de commande	Création d'une commande avec attributs. Interaction avec les devis, la gestion des stocks, la livraison. Génération de lignes de commandes, total commandes. Historisation des commandes	Lien obligatoire avec le dossier, le contact, référentiel produits. Lien potentiel avec les devis, stock, livraison. Ne peut concerner que les dossiers de nature commerciales.												
	Facturation	Lié à une réf. (implication comptable) Edition de facture	Edition de facture basée sur la commande, ou sur saisie libre. On peut générer de la facturation à partir des feuilles de temps. Peut être liée au workflow de l'activité. Les modèles de factures sont utilisés depuis la GED (multilingue possible)	Ne concerne pas les associations, le marketing												

 Gestion financière	Remises (chèques / versements)	Possibilité de constituer des remises de chèques ou virements en y associant plusieurs factures. L'application conserve l'historique des remises déposées ou à valider	Saisie de remises de chèques ou de virements sur les factures correspondantes. Historisation	Lien facturation											
	Rapprochement des charges et des produits														
	Rapprochement bancaire ou comptabilité	Rapprochement des encaissements réels avec des données bancaires													
 Gestion des documents	Classement des modèles	Gestion simple des documents (classement, attachement)	Attachement d'un document de la GED, d'un document entrant (email), d'un document local au dossier, activité. Classement selon process métier												
	GED (versioning, indexation...)	Gestion électronique de documents avancée	Intégration de documents existants dans une bibliothèque (sans relation), attachement à n dossiers, n activités, n ressources (avec lien relationnel). Lecture et modification des documents directement dans l'outil (tout ou partie du document, fusion de documents entre eux).												
	Découpage et gestion de pièces	Découpage de documents PDF en plusieurs extraits, numérotation des parties, affichage distinct des parties, ...	Extraction de tout ou partie d'un document PDF en X morceaux, avec renommage automatique selon nomenclature métier. Numérotation des parties, tampon sur document, affichage par partie. Attachement à un dossier et à une activité, classement dans la GED												
 Pilotage	Rapporting d'activité (tableaux standards)	Mise en place d'indicateurs métier de base, mise en page dans des rapports standards.	calcul d'indicateurs standards métier (cf. expression des besoins) et mise en page classique. Pas d'interaction	CA temps passé, calcul par ressource, par dossier, etc...											
	Gestion des prévisions	Mesure d'écart entre les objectifs liés à un dossier métier -ou autre objet- et le réel	Soustraction entre temps passé et charge déclarée (feuille de temps), budget dépensé et budget réel (facturation), ...												
	Analyse des risques et qualité	Evaluation notée et/ou pondérée d'un objet	Evaluation d'un dossier : Notion de priorisation pour les projets par ex. (surtout projets potentiels) gestion des risques (de base : sur simple notation, ou sur algorithme liant budget, temps, FH, etc.). Impact sur les dossiers entre eux	Evaluation de ressources (performance), évaluation du risque (projet)											
 Gestion des Ressources humaines	Gestion des Ressources Humaines	Gestion des ressources internes ou externes qui peuvent être affectées à un dossier : qualification, compétence anonyme ou nommée, lien avec les ressources	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Interaction avec les activités (affectation), l'agenda (disponibilité), la planification (simulation), la gestion des revenus (charges financières)	Les ressources humaines peuvent être incluses dans la table des contacts (gestion des rôles)											
	Admin. Interne (congrés, notes de frais)	Gestion des contraintes internes liées aux ressources et aux dossiers.	Saisie des congés dans les plannings. Rattachement aux dossiers, aux contacts	Saisie des notes de frais. Rattachement aux dossiers, au contacts. Interaction avec la facturation (charges financières).											
 Gestion des Ressources matérielles et immatérielles	Gestion des Ressources Matérielles/Immatérielles	Gestion des ressources matérielles et immatérielles, internes ou externes qui peuvent être affectées à un dossier : qualification, Sélection, affectation	Consultation, modification, ajout, suppression d'enregistrements et qualification de leurs attributs (saisie libre, ou menus déroulants issus de listes ou d'objets de la base). Interaction avec les activités (affectation), la gestion des revenus (charges financières)	Les ressources matérielles et immatérielles sont liées à des référentiels "produits"											
	Lien Messagerie	Gestion du canal écrit (email) en one-to-one	Ensemble des notifications, alertes, mails liés aux activités sont gérées par la messagerie de l'utilisateur (lien sur messagerie externe, pas de messagerie interne)												
 Gestion des canaux de communication	Multicanal	Gestion du canal écrit (email et SMS) en masse	Capacité à diffuser en masse des messages écrits à des contacts externes selon les principaux canaux de communication (emails, SMS). Cela peut être un service dissocié de l'outil ou intégré. Les fonctions principales sont le routage selon des règles métier et la traçabilité (réactions des contacts aux messages).												
	Collaboratif	Espace de collaboration et d'échanges	Espace de discussion, Tchat ou forum, utilisable inter-utilisateurs												
	CTI	Gestion du canal voix, rattachée à l'outil.	Remontée automatique de fiche contacts, liaison à l'application d'un centre d'appels (lien avec la base de données pour reconnaissance clients entrants, gestion de base des scripts (discours), saisie des données de retours												
	Import de données	Mise à jour de données récurrentes	Module de base d'import de données simples, avec dessin d'enregistrement préparamétré (données récurrentes : contacts, produits)	Catalogue produit, cible marketing (location fichier)											

 Gestion des interfaces	Export de données	Export de données dans un format standard	Dessin d'enregistrement à prévoir pour permettre de fournir les données incluses dans l'outil												
	Export de documents	Mise à disposition de documents	Export d'un dossier de document avec génération automatique de sommaire												
	Mobilité Synchronisation	Mise à jour des données sur appareil mobile	Accès à l'application sur appareil mobile, synchronisation des applications locales												
	Echange avec tiers / Partenariat (e-barreau, SAGE par ex.)	Gestion des entrées et sorties de données depuis et vers l'externe	Envoi ou mise à disposition, voire échanges de données vers l'extérieur (fichier emailing vers routeur, greffe, ...) . Problématique de cryptage												
 Gestion des produits et services	Nomenclature Produit/service	Table des produits et services													
	Stocks	Calcul des stocks en fonction des produits écoulés	Gestion des stocks de produits en fonction des commandes (stock initial - produits écoulés), alertes sur état de stock A gérer : Bons de Livraison												
	Livraisons	Evaluations des coûts, disponibilité pour le transport de marchandise	Gestion des coûts de livraison en fonction du référentiel produit (poids, taille) et des données géographiques de la table clients; action vers un tiers (fournisseur livraison)												
	Boutique	Gestion des éléments liés au paiement et à la facturation, à la reconnaissance client, à l'historisation de la vie du client et de son comportement	Gestion du paiement en ligne, sécurisation, reconnaissance utilisateur (rappel des comptes personnels), suivi des états de commandes (achats, état des livraisons...), mise à disposition des factures (générées depuis les commandes), push d'offres sur le site en fonction de scénarios...												
	Cartographie	Localisation géographique ou visualisation de données brutes ou calculées sur un support géographique.	Calculs de distance (activités), placement de lieux sur une carte (dossier ou activité), liaison adresse avec dossier (activité ?). Géolocalisation.	Dossiers de nature commerciale (activité de type FM) par ex. Marketing : ciblage selon position géographique (géomarketing)											
 Gestion de Support technique	Gestion de demandes d'assistance ou d'intervention technique.	Centralisation et ordonnancement des demandes de service.	Qualification de la demande (attributs pré-paramétrés), gestion par ticket (N°identifiant), génération d'alertes, notification et de récapitulatifs par mail, lien avec les contacts (rôle). La demande peut devenir un dossier selon les règles métiers prédéfinies.	Dossiers de nature Projet essentiellement											
	Datamining (service)	En mode Service bureau, effectué par l'inside de façon ad hoc.	Analyse de données selon des modèles statistiques complexes. Etudes Quantitatives ad hoc.	Requiert de manipuler des données cible en dehors de la base native pour les soumettre à des algorithmes statistiques spécifiques.											
 Analyses	Requêteur	Sélection de cible (en mode listing) ou comptage avec tous les opérateurs possibles	Le requêteur peut attaquer l'ensemble des attributs (caractéristiques) de la base (contacts, produits par exemple), et utilise des opérateurs multiples et complexes (Et, Ou, Inclusion, Exclusion...). Les résultats (comptage) de chaque requête est gardée. La requête est construite étape par étape sans perte des sous requêtes précédentes. Le résultat est au format liste (enregistrements définis par N°identifiant) et peut s'enregistrer. Le format d'édition de la liste cible peut être paramétré et exportable.	Individus, produits, dossiers, activités (selon statut). Pour le marketing par ex. sélection de cible (adéquation offre/ cible) Balance financière...											
	Business Intelligence (tableaux complexes)	Création d'analyse spécifique par utilisateur avec mise en page personnalisée	Création par l'utilisateur ou par l'inside de nouveaux indicateurs non disponibles en standard selon les données disponibles en base. Mise en page personnalisée. Analyse multidimensionnelle	Besoin spécifique métier, mise à disposition "d'univers" ou de données agrégées.											

II. Suivi projet

L'outil de gestion de projet nous permet de suivre la planification de chaque demande et la liaison entre chacun d'elles. Il assure également le travail collaboratif avec le processus de validation des développements effectués.



III. Architecture du système d'information

Le système d'information liée à la nouvelle technologie choisie s'organise en partie de la façon suivante. Ce schéma a été réalisé pour expliciter au comité de pilotage l'architecture retenue.

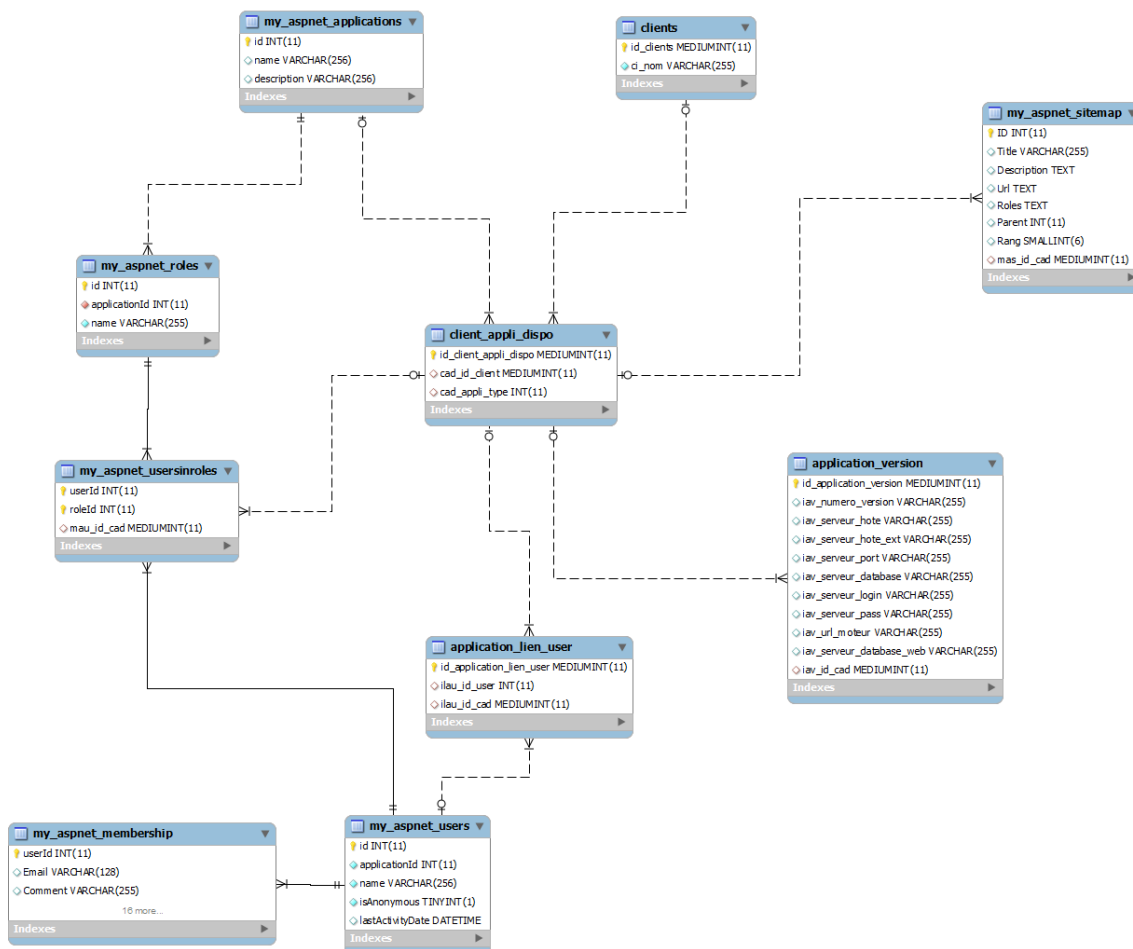
Serveur			
Windows serveur 2003/2008			
Outils			
IIS 7.0	PHP/MySQL	ASP.net	Serveur FTP
BDD			
BDD INSIIDE	BDD Produit	BDD Intranet	
Applications / Sites Web			
Insiide Admin	Produit V1	Produit V2	Intranet
Webservices			
WS Insiide Admin	WS Produit V1	WS Produit V2	

IV. Modèle conceptuel de données (MCD)

Le modèle conceptuel de données (MCD) présenté correspond à la nouvelle base de données créée afin d'intégrer une authentification unique pour l'ensemble des applications. Nous avons mis au centre de la réflexion la table « client_appli_dispo » qui est le point d'entrée, qui nous permet de définir quel client à accès à quelle application.

Nous gérons également une notion de version d'application et de base de données liés qui nous permet d'avoir pour chaque application la base correspondante. Cela est utile si un client conserve une ancienne version alors qu'un autre a souhaité avoir une mise à jour majeure.

Un autre point traité en amont est la gestion du menu pour les applications. Qu'elles soient client léger ou client riche, la gestion du menu dynamique est une fonctionnalité qui diminue les opérations de maintenance du point de vue du développement.



I. MCD Nouveau produit

Le modèle conceptuel de données du nouveau produit est donc composé de trois parties :

- le MCD commun à toutes les applications (présenté ci-dessus),
- le MCD spécifique « association »,
- le MCD « générique », c'est-à-dire des tables utilisées par tous les logiciels de la suite.

