



HAL
open science

Système de suivi des problèmes (Bugs) : B-WARN Suite

Pierre El Khoury

► **To cite this version:**

Pierre El Khoury. Système de suivi des problèmes (Bugs) : B-WARN Suite . Réseaux et télécommunications [cs.NI]. 2011. dumas-01103941

HAL Id: dumas-01103941

<https://dumas.ccsd.cnrs.fr/dumas-01103941>

Submitted on 15 Jan 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 METIERS

**CENTRE REGIONAL ASSOCIE DE
BEYROUTH**

MEMOIRE

présenté en vue d'obtenir

le DIPLOME d'INGENIEUR CNAM

SPECIALITE : Informatique

OPTION : Réseaux, Systèmes et Multimédia (IRSM)

par

EL KHOURY, Pierre

Système de suivi des problèmes (Bugs)

B-WARN SUITE

Soutenu le

13 Avril 2011

JURY

PRESIDENT : Pr. Jean-Pierre ARNAUD

MEMBRES : Ing. Pascal FARES

Dr. Amal KOBEISSI

M. Antoine NAJEM

M. Johny BECHARA

Remerciements

Mes premiers remerciements s'adressent à Monsieur Antoine NAJEM, propriétaire de PIXEL, pour m'avoir accordé l'opportunité de développer mes compétences professionnelles au sein de son entreprise, et son accord à présenter ce projet comme sujet de mon mémoire d'ingénieur.

Je remercie également Madame Amal KOBEISSI, ma tutrice, pour son aide, sa patience, et tout le temps qu'elle m'a consacré, tout au long du développement de ce projet, et de la rédaction de mon mémoire.

Je remercie Monsieur Pascal FARES, responsable de la filière informatique à l'ISAE, centre régional de Beyrouth, et tous les membres de l'équipe d'enseignement, pour leurs efforts qui m'ont permis de suivre mes études supérieures parallèlement avec la vie active.

Je remercie aussi tous les membres du jury, dont Professeur Jean-Pierre ARNAUD qui me fait honneur de le présider.

Je remercie Monsieur Elie ABOU RJEILI, mon professeur de langue française, pour son assistance et son aide appréciable durant la rédaction de ce mémoire.

Une pensée très particulière va à ma petite famille, mes parents Elias et Siham EL KHOURY, ma sœur Diala et son époux Joe FADEL, pour leur soutien total pendant tant d'années durant lesquelles j'ai suivi les cours du soir, et leur encouragement tout au long de la préparation de mon mémoire.

Enfin, à tous mes collègues à PIXEL, ma deuxième famille, j'adresse une salve de remerciements pour leurs contacts chaleureux, et leur esprit de fraternité qui a adouci mes dures journées de travail.

Glossaire

Acronymes

AJAX	Asynchronous JavaScript And XML
CRM	Customer Relationship Management
CTI	Computer Telephony Integration
DFLGLG	Délégation générale à la langue française et aux langues de France
GRC	Gestion Relation Client
HA	High Availability
HTML	Hyper Text Markup Language
IIS	Internet Information Services
ISP	Internet Service Provider
ITIL	Information Technology Infrastructure Library
MTS	Muli-Tiered Support
PDA	Personal Digital Assistant
PME	Petites et Moyennes Entreprises
QA	Quality Assurance
RTF	Rich Text Format
SGBD	Système de Gestion de Base de Données
SLA	Service Level Agreement
UML	Unified Modeling Language
VB	Visual Basic
VSS	Microsoft SourceSafe
WCF	Windows Communication Foundation
WPF	Windows Presentation Foundation
WWF	Windows Workflow Foundation

Définitions

- **Métrologie**

Technique qui permet d'évaluer et de mesurer les performances d'un système informatique (1).

- **Obfuscation / Obscurcissement**

Opération qui consiste à transformer une section de code ou un programme, de manière à le rendre totalement incompréhensible à un lecteur humain, même aidé d'outils informatiques (1).

- **Filigranage**

Technique de marquage qui consiste à insérer une signature invisible et permanente à l'intérieur des images numériques transitant par les réseaux, tel Internet, afin de lutter contre la fraude et le piratage et d'assurer la protection des droits de propriété intellectuelle (1).

- **Rétro-ingénierie**

Pratique qui consiste à analyser un produit fini (comme un logiciel d'application ou une puce) pour connaître la manière dont celui-ci a été conçu ou fabriqué (1).

Sommaire

<i>Introduction</i>	7
<i>Première partie : Environnement du travail</i>	10
Chapitre I : L'entreprise.....	11
1. Produits et services.....	11
2. Structure de l'entreprise.....	12
3. Historique.....	15
Chapitre II : Problématique.....	18
1. Situation.....	18
2. Identification des besoins.....	19
3. Dégagement des objectifs.....	20
<i>Deuxième partie : Etude bibliographique</i>	23
Chapitre III : Gestion de la relation client.....	24
1. Définition.....	24
2. Marketing transactionnel et Marketing relationnel.....	25
3. Multi-canal de communication.....	26
4. Avantage du GRC.....	27
Chapitre IV : Service client.....	29
1. Définition.....	29
2. « 5 P's » du service client.....	30
3. Service à Multi-tiers.....	31
4. Le coût d'un mauvais service.....	33
Chapitre V : Systèmes de Suivi et d'Assistance.....	35
1. Définition.....	35
1.1 Système de suivi des problèmes.....	35
1.2 Système de gestion du service d'assistance.....	36
2. Principe des Tickets d'incident.....	37
3. Gestion de priorité des bugs.....	37
4. Cycle de vie d'un bug.....	40
Chapitre VI : Solutions disponibles.....	42
1. Enquête et critères.....	42
2. Etude comparative.....	43
2.1 Catégories des systèmes.....	44
2.2 Types des applications.....	44
2.3 Canaux de communications.....	45
2.4 Fonctions intégrées.....	46
3. Portée des recherches.....	48

<i>Troisième partie : Solution proposée</i>	50
Chapitre VII : Mise en œuvre	51
1. Description de la demande	51
1.1 Objectifs.....	51
1.2 Produit du projet	52
2. Choix des technologies.....	55
3. Cycle de vie et planification.....	58
4. Equipe et répartition des tâches.....	61
Chapitre VIII : B-WARN SUITE	62
1. Paramètres généraux	62
2. Interprétation des composants	65
2.1 B-WARN	65
2.2 LIVE B-WARN	84
3. Interprétation technique.....	87
3.1 La base de données	87
3.2 B-WARN	88
3.3 LIVE B-WARN	92
4. Implémentation	93
Chapitre IX : Conduite du projet.....	96
1. Problèmes rencontrés	96
1.1 Problèmes d'ordre organisationnels	96
1.2 Problèmes de conception	99
1.3 Problèmes techniques.....	102
2. Plan appliqué.....	104
3. Evaluation	106
<i>Conclusion</i>	109
<i>Annexes</i>	112
Annexe 1 : Le premier bug informatique	113
Annexe 2 : Des bugs informatiques catastrophiques.....	115
Annexe 3 : La fonction stockée fnGefTicketAssignedTo(int,date).....	117
Annexe 4 : Le déclencheur Trig_TK_Tickets_Insert	118
Annexe 5 : Classe Projects du projet SYS.DataAccess.....	119
Annexe 6 : Classe ServiceLevels du projet SYS.BusinessRules.....	121
<i>Bibliographie</i>	123
<i>Liste des figures</i>	125
<i>Liste des tableaux</i>	126

Introduction

PIXEL, mon lieu de travail actuel, est une entreprise qui offre des produits ou logiciels informatiques qui traitent plusieurs fonctions comme la gestion du stock, la gestion financière et surtout la gestion des différents types de l'affaire d'assurance : courtage d'assurance, assurance, réassurance... Son champ de service couvre principalement le Liban et plusieurs pays arabes comme la Syrie, l'Iraq, l'Arabie Saoudite et les Emirats Unis.

Elle a débuté par un effort personnel, et elle emploie actuellement une dizaine de personnes techniquement très bien qualifiés, responsables d'offrir aux clients les meilleures solutions de gestion possibles. En plus, le personnel de l'entreprise est responsable du service client qui comporte des mises à jour de leurs versions des logiciels et du développement des interfaces, modules ou rapports supplémentaires exigés par leurs besoins. Ce personnel est relativement petit par rapport aux services offerts par PIXEL, ce qui a conduit à ce que certains membres prennent en charge plusieurs responsabilités en même temps.

Pendant les derniers mois de l'an 2008, le département du service client PIXEL était dans une situation de désordre total, ce qui affectait sérieusement la performance de son personnel. Cette situation était due à l'accroissement immense de son portefeuille qui était le résultat des efforts énormes de l'équipe de travail. Les membres de cette équipe ont réussi à produire des versions de logiciels intégrant des nouvelles fonctionnalités, qui ont fasciné plusieurs sociétés d'assurance, ce qui a augmenté la demande d'achat et par la suite les requêtes et les tâches du service client.

L'administration de PIXEL se trouvait confortée à une situation critique qui menaçait la continuité de l'entreprise. Plusieurs objectifs ont été fixés afin de résoudre cette situation. Le développement d'un logiciel GRC (Gestion de la relation client) qui réunit toutes les fonctions de traitement et de suivi des problèmes et des requêtes des clients, était un des ces objectifs.

Ce type de logiciel nécessite un temps énorme de développement, ce qui n'était pas réalisable, vu les surcharges et le manque de personnel. La solution étant urgente, l'administration a décidé de développer un système de suivi des tickets d'incident qui sera amélioré ultérieurement par un logiciel GRC. C'était la décision du développement de B-WARN SUITE, sujet de ce mémoire.

Au début de ce travail, j'ai été nommée chef d'une équipe formée de quatre personnes, responsable du développement de ce projet. Pendant cette période, j'ai été responsable de la planification des tâches, de leurs répartitions sur les membres de l'équipe, et de la supervision du développement, afin d'atteindre les objectifs fixés à ce projet.

Après un travail d'équipe qui a duré trois mois, et en raison de la surcharge des tâches et du manque de personnel, un changement de priorité a été adopté, ce qui a conduit à ajourner le développement de B-WARN SUITE. Mais après plusieurs discussions, j'ai pris en charge le développement du reste du projet, qui en représentait approximativement la moitié, avec une minime assistance des autres membres durant les phases de tests. Au cours de cette période, j'ai été responsable de la conception et du codage de tous les composants et toutes les fonctions restantes.

Les différentes phases et étapes de la vie de ce projet sont décrites dans ce document, il comporte trois parties :

La première partie intitulée « Environnement du travail », contient deux chapitres. Le premier définit l'entreprise en fonction des services offerts et de la répartition des responsabilités. Ce chapitre comprend aussi l'historique de l'entreprise dès sa fondation jusqu'à présent. Le deuxième chapitre est concentré sur la situation qui a engendré le développement de B-WARN SUITE. Il décrit en détail les différents problèmes de cette période et tous les besoins exigés.

La deuxième partie, « Etude bibliographique », comporte plusieurs chapitres traitant différents points en rapport avec le sujet projet. Ces chapitres comportent une définition de la GRC et ses avantages, et du service client et l'importance de l'architecture du service à multi-tiers. En plus, ils établissent une comparaison entre les systèmes de gestion de la relation client et les systèmes de suivi des problèmes (ou incidents), qui sert à catégoriser le projet et à identifier ses fonctions essentielles. De même, le concept de ticket de suivi et le cycle de vie d'un ticket sont exposés dans ces chapitres, pour mieux comprendre leur importance et les intégrer dans B-WARN SUITE.

A la fin de la deuxième partie, un chapitre exposera les résultats de l'enquête réalisée sur les produits existants sur le marché qui traitent le même sujet que ce projet, et les mêmes fonctionnalités souhaitées. Ce chapitre aide principalement à prendre conscience de l'état

actuel de ces types de logiciels, comme il aide à découvrir les fonctions et les innovations incorporées dans chacun.

La troisième partie de ce document nommé « Solution proposée », comporte deux chapitres. Le premier est destiné à décrire les objectifs du projet et les étapes de sa mise en œuvre, comme il comporte une description des technologies utilisées, la planification initiale du projet et son cycle de vie. Dans le deuxième chapitre, les différents composants de B-WARN SUITE sont détaillés et toutes leurs fonctionnalités intégrées. De même ce chapitre comporte une description détaillée des pré-requis nécessaires à l'implémentation du projet, et une description des différentes méthodes d'installation valables. A la fin de cette partie, un chapitre décrira tous les problèmes rencontrés au cours du développement du projet, et leurs impacts sur le déroulement du travail et sur le respect de la planification initiale.

Enfin, une conclusion terminera ce rapport, elle intègre des perspectives concernant le projet et ces fonctionnalités, et intègre aussi un bilan personnel évaluant l'expérience personnelle acquise au cours du développement du projet, et après son achèvement.

Première partie

Environnement du travail

Avant de commencer, il serait important de définir l'environnement de travail dans lequel ce projet a eu lieu. Cette partie comporte une présentation de l'entreprise, de ses services et de son historique et une description de mon évolution professionnelle au sein de son équipe. Ensuite, sera exposée la situation de l'entreprise qui a engendrée le projet, sujet de ce mémoire.

Chapitre I : L'entreprise

Au cours de ce chapitre, l'entreprise sera définie en fonctions des services offerts, de l'équipe de travail, de ses qualifications, ses compétences, sa structure, son historique et sa vision, et de ma progression dans son équipe.

1. Produits et services

PIXEL propose à ses clients des produits (logiciels) modulaires développés sous le Framework .NET offert par Microsoft, opérant sous le système d'exploitation Windows.

Ces produits constituent des solutions complètes qui couvrent tous les champs d'affaires des clients. Ils débutent par la gestion des contacts et de toutes les données nécessaires des acteurs internes et externes du travail du client, et se termine par la gestion financières des comptes et des transactions comptables en passant par toutes les fonctions nécessaires pour chaque type d'affaires, qui assurent la meilleure conduite du flux du travail.

En outre, l'entreprise offre un service d'assistance et de mise à jour continuels de ces produits afin d'assurer le bon fonctionnement et la continuité de ces derniers.

Son champ d'opérations couvre principalement le Liban et s'étend à la Syrie, l'Iraq et quelques pays du Golf, notamment l'Arabie Saoudite, les Emirats Arabes Unis et autres.

La majorité des clients de PIXEL sont des acteurs du domaine d'assurance : agents de courtage, sociétés d'assurance, sociétés de réassurance... Ces entreprises utilisent IRIS.Net *Insurance and Reinsurance Information System* développé par PIXEL en plusieurs modèles et versions pour couvrir tous les types d'affaires du domaine d'assurance.

IRIS est composé de plusieurs modules qui couvrent tous les détails de ce vaste domaine d'affaires, tels que les modules de gestion des contacts, de souscription des polices d'assurance, de gestion des accidents... ainsi que d'autres modules : celui de gestion des accès, de gestion financière et les rapports statistiques qui s'ajoutent à IRIS pour former, tous ensemble, un système de gestion complet, sécurisé et rigide.

IRIS.Pos (Point of Sale), une application web optionnelle complémentaire qui peut être associée à IRIS.Net, offre aux clients la possibilité de relier tous leurs points de vente à la

même base de données centralisées, ce qui assure la cohérence des données et la possibilité de maîtrise instantanée des ces données.

Un autre produit de PIXEL : Financial.NET. Il est conçu pour gérer la fabrication et la production de tout genre de produits, le traitement des salaires, en plus de la gestion de vente et, implicitement, celle du stock.

Une version spéciale de ce logiciel était développée afin de fonctionner sur les assistants numériques personnels PDA. Cette version qui peut être considérée comme complémentaire du module de vente, permet la communication directe entre les représentants de commerce et la base de données de leur société.

De même, Financial.NET incorpore les modules de gestion des accès, de gestion financière et les rapports des statistiques.

Outre les produits cités ci-dessus, PIXEL offre à ses clients d'autres modules considérés comme des outils de management qui s'intègrent dans tous ces logiciels. Ces modules, comme le Business Intelligence, les services de notifications et la gestion des autorisations, offrent aux logiciels un facteur concurrent sur le marché susceptible d'attirer en plus les gestionnaires des sociétés.

Par ailleurs, plusieurs interfaces de communications sont intégrées dans les logiciels de PIXEL. Ces interfaces servent à récupérer (ou envoyer) les données d'autres logiciels des partenaires des clients, basés dans les pluparts des cas sur des différentes plate-formes et à différents systèmes de gestion des bases de données (SGBD).

2. Structure de l'entreprise

L'équipe de travail de PIXEL est relativement petite par rapport à son portefeuille et au nombre de ses clients. Cette équipe regroupe des débutants et des professionnels techniquement bien qualifiés. Elle est unie par un esprit familial qui, aiguisé par les ambitions individuelles, constitue la motivation des employés à offrir à leur entreprise tous les efforts nécessaires à favoriser sa réussite dans ces différents projets.

Les employés de PIXEL sont adaptés au développement des logiciels à l'aide des langages de programmation VB.Net, ASP.Net offerts par Microsoft. De même, l'équipe

utilise Microsoft SQL Server pour créer et gérer non pas seulement les bases de données des logiciels produits, mais aussi tous leurs composants comme le *Reporting Services* et le *Notifications Services*.

En intégrant dans les logiciels des interfaces de communication et de synchronisation avec d'autres logiciels, les membres de l'équipe ont étendu leurs compétences et leur connaissance afin de couvrir d'autres SGBD comme ORACLE, MySQL et autres.

Afin de couvrir tous ces champs de services (développement de logiciels et services d'assistance) qui ont nettement évolué dans les trois dernières années, l'administration de l'entreprise a élaborée deux structures indépendantes pour répartir les fonctions et les responsabilités du personnel. La première s'applique au cours du développement de nouvelles tâches et projets, et la deuxième pendant le service d'assistance des clients assuré après la vente des produits.

La structure utilisée au cours du développement (Figure 1) consiste à diviser le personnel en deux équipes, guidées chacune par un chef d'équipe. Chaque équipe est responsable du développement des tâches de l'un des produits de PIXEL (IRIS.Net et Financial.NET). Ces équipes sont administrées par le directeur de développement et son assistant qui assurent la communication avec l'administration de l'entreprise.

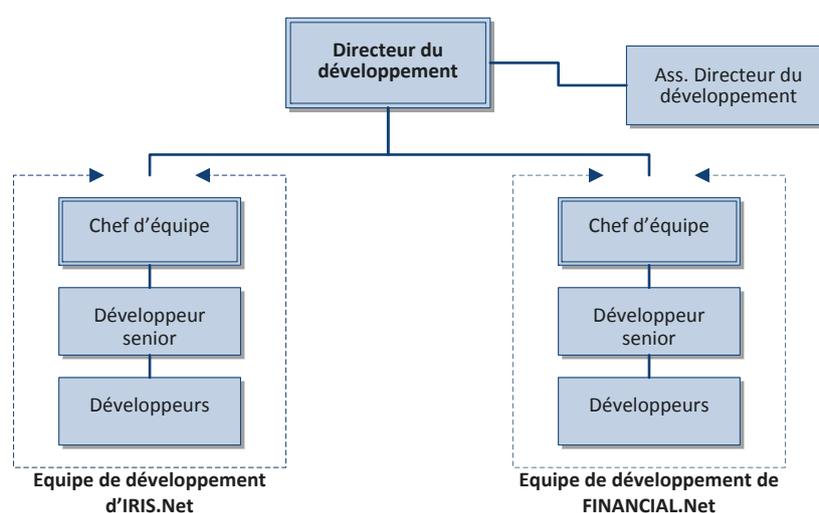


Figure 1 : Structure organisationnelle utilisée au cours du développement.

La structure utilisée pendant le service après vente (Figure 2) regroupe le personnel en trois niveaux de service selon l'expérience professionnelle des membres : Helpdesk, Service

technique et Service avancé. Cet arrangement assure le meilleur service client selon les catégories des problèmes qui seront identifiés par le directeur du service et assignés au niveau du service convenable.

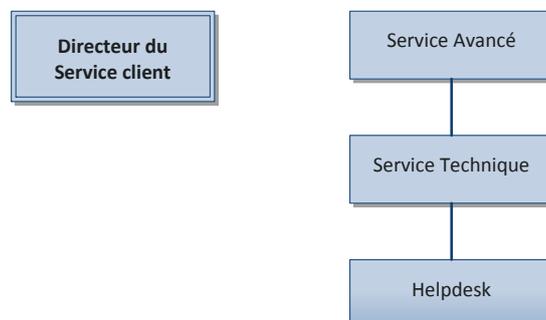


Figure 2 : Structure organisationnelle utilisée au cours du service après vente.

La majorité du temps, ces deux structures seront appliquées en parallèle, ce qui nécessite une coopération entre les deux directeurs de développement et du service d'assistance, afin de diminuer les surcharges des membres qui participent aux deux structures.

En plus de ces deux structures, un *Account Executive* sera désigné à chaque nouveau projet afin d'en assurer le suivi, dès la phase de sa conception jusqu'à la fin de sa phase de mise en service. Il sera responsable d'assurer, en accord avec le directeur de développement, le développement des tâches conformément aux exigences du client, et de reporter la progression du projet à l'administration de l'entreprise.

Enfin, un directeur technique est responsable de toutes les recherches nécessaires aux évolutions des technologies utilisées par l'entreprise et de la formation nécessaire du personnel, afin de pouvoir les mettre en application.

Ces structures organisationnelles utilisées par PIXEL obligent certains employés à se charger de deux et parfois de trois fonctions en même temps, ce qui nécessite des efforts exceptionnels pour achever toutes leurs responsabilités assignées.

Cela a été la meilleure solution temporaire trouvée par l'administration pour couvrir tous les services offerts par PIXEL, jusqu'à pouvoir assurer le personnel nécessaire et aboutir à une structure unifiée.

3. Historique

PIXEL est une société à responsabilité limitée (SARL), fondée en 1996 par son associé principal et directeur actuel M. NAJEM Antoine. Elle a débutée par un effort individuel pour regrouper de nos jours une dizaine de personnes.

Pendant ses premiers 8 ans, PIXEL était une petite entreprise qui vendait des logiciels de comptabilité, de gestion de stock et de gestion de courtage d'assurance. Ces logiciels traditionnels étaient développés sur Microsoft Visual FoxPro.

En 2004, les produits de PIXEL sont redéveloppés en utilisant Microsoft Visual Studio 2003. Pendant cette période, l'équipe de PIXEL comportait 5 personnes.

Au milieu de l'année 2006, durant la guerre de juillet au Liban, l'équipe de PIXEL a profité du temps mort pour développer la deuxième version d'IRIS, en utilisant les logiciels récents de Microsoft, Visual Studio 2005 et SQL Server 2005.

C'est à ce temps-ci que j'ai débuté ma carrière à PIXEL en tant que stagiaire, ce qui m'a donné l'occasion de participer aux réunions d'analyse et de conception de cette nouvelle version, et de devenir, deux mois plus tard, employé et membre de l'équipe de développement d'IRIS.

La nouvelle version d'IRIS a été adoptée par deux sociétés d'assurances locales de niveau moyen, comportant environ cinquante personnes chacune, juste après la fin du développement.

A cette période, j'ai participé au service après vente du nouveau produit et j'ai été chargé de réaliser des recherches sur deux jeunes composants de Microsoft le *Business Intelligence* et les *Notifications Services*, pour exploiter leurs fonctionnalités et informer les membres de l'équipe de ces nouveautés, afin de les intégrer dans les logiciels développés par PIXEL.

Au bout de ces recherches, IRIS.Pos, l'application web complémentaire d'IRIS, a été commandée par un de nos clients. A cette période, aucun membre n'avait une expérience dans le développement des applications web : toutes nos connaissances concernant ce type d'application étaient basiques et théoriques. C'était mon premier grand défi à PIXEL : gérer le développement de la première application web, sans aucune assistance. Cette application

était développée en ASP.NET, et en utilisant la technologie AJAX (Asynchronous JavaScript And XML).

Pendant le deuxième semestre de l'an 2007, l'administration de PIXEL a préparé trois propositions pour trois offres de développement de logiciels au profit de grandes sociétés (comportant plus de 100 personnes chacune), espérant que l'une d'elles sera acceptée. A notre grand étonnement, toutes ces trois propositions ont été acceptées ! C'était pour toute l'équipe un véritable challenge de pouvoir réussir à développer ces projets, en respectant les dates limites.

Vu le manque du personnel, j'ai été chargé d'assister un des développeurs âgé dans la conception de l'un des trois nouveaux projets, appelé IRIS-Re. C'était un système de gestion de courtage de réassurance : un type d'affaire neuf pour tous les membres de PIXEL. Quelques mois plus tard, et pour les mêmes raisons, je suis devenu le responsable du suivi du développement d'IRIS-Re.

L'année 2008 représente la période de propagation à l'étranger. Les nouvelles versions d'IRIS déjà développées ont satisfait les sociétés d'assurance dont elles se sont pourvues. Cet impact s'est propagé dans le consortium du domaine d'assurance, ce qui a amplifié les demandes d'achat des différentes versions du logiciel. C'est le début d'une nouvelle ère à PIXEL : celle de l'évolution et du professionnalisme.

Grace à cette évolution, l'entreprise a obtenu la certification ISO 9001:2008 *Systèmes de management de la qualité-Exigences*, signe de début de sa marche professionnelle.

Durant le dernier trimestre de l'an 2008, j'ai été chargé de la mise en place d'IRIS en Syrie et plus tard à Abu-Dhabi. Telles sont mes premières expériences professionnelles à l'étranger.

Au début de l'an 2009, étant donné de la surcharge permanente des tâches de développement et de services, une restructuration du personnel est réalisée. Cette nouvelle organisation me promeut la position de directeur du service d'assistance clientèle. Mais à cause de l'émigration inattendue du membre le plus âgé et le plus qualifié de l'équipe, j'ai été libéré de ma nouvelle position pour assumer la totale responsabilité de deux mises en place consécutives, dans deux grandes sociétés internationales situées à Dubaï, de la troisième

version d'IRIS développée en utilisant Visual Studio 2008 et SQL Server 2008. La dernière de ces mises en place a été achevée au cours des semaines de la rédaction de ce mémoire.

De nos jours, PIXEL emploie 14 personnes dont l'expérience professionnelle varie entre un et cinq ans. Elle est en état de recrutement continu de personnel afin de couvrir tous ses services offerts et les besoins de ses clients. Ce recrutement se réalise d'une manière lente et vigilante, de sorte que seront maintenus le même niveau de compétence du personnel et le même esprit d'équipe.

Il serait judicieux de noter que l'entreprise se prépare à une évolution vers les technologies récentes du .NET Framework comme le WCF (Windows Communication Foundation) le WPF (Windows Presentation Foundation). Egaleme nt, l'équipe est en cours de développer plusieurs logiciels qui seront mis en place entre la fin de l'an 2010 et le début de l'an 2011. En plus, PIXEL s'occupe des services de maintenance et d'assistance des produits déjà vendus, et de toutes les fonctions de mise à jour nécessaires aux clients.

Aujourd'hui, en ma qualité d'*Account Executive*, je suis chargé du suivi des deux projets récemment mis en place à Dubaï. De même, comme *Développeur Senior*, je suis le responsable de la conception et du développement des modules de réassurance intégrés dans les différentes version d'IRIS, et je participe au niveau *Service Avancé* du service client pour les mêmes modules.

Chapitre II : Problématique

Le développement du projet, sujet de ce mémoire, n'est pas arbitraire mais défini par l'administration, pour faire partie intégrante de la solution susceptible de combattre le désordre et la rivalité, engendrés par l'accroissement considérable du portefeuille de l'entreprise.

Ainsi, au cours de ce chapitre, seront exposés les différents problèmes et complications rencontrés, représentant à priori la raison initiale de la naissance du projet.

1. Situation

A la fin de la première moitié de l'an 2006, toutes les versions d'IRIS développées étaient basées sur une nouvelle culture de conception adaptée par PIXEL. Cette culture consiste à développer des logiciels dynamiques et complètement personnalisables selon les besoins des clients sans aucune modification du code, à l'exception des besoins spécifiques qui diffèrent un client d'un autre.

De plus, des nouvelles fonctionnalités ont été intégrées dans les logiciels récemment créés, telles que les fonctions d'autorisations qui permettent de limiter les permis des utilisateurs et de gérer les demandes d'autorisation pour les différentes fonctions des systèmes et les services de notification qui servent à notifier les responsables (par email ou SMS) des opérations critiques.

Ces nouveautés de PIXEL ont fasciné plusieurs sociétés d'assurance, ce qui a multiplié les demandes d'achat des différentes versions d'IRIS.

L'accroissement inattendu et non organisé du portefeuille de PIXEL a engendré une situation de désordre incontrôlée, ce qui a sérieusement affecté la performance de l'équipe et par le fait même la satisfaction des clients. Cette situation atteint son apogée à la fin de l'an 2008. L'administration de PIXEL se trouve confortée face à des problèmes critiques qui menaçaient la continuité de l'entreprise et nécessitaient une solution complète et une réorganisation de tous les services et du personnel de l'entreprise.

2. Identification des besoins

Durant cette période, plusieurs changements sont imposés sur l'environnement du travail. Ces changements ont exigé de nouveaux besoins pressants et nécessaires à la continuité de l'entreprise.

Dans ce qui suit une description des changements majeurs de l'environnement de travail et des besoins dictés par chacun d'eux.

a. L'allure des nouveaux clients

Formant des moyennes ou grandes entreprises (comportant jusqu'à 100 employés) bien organisées et professionnelles, les nouveaux clients ont adopté des nouvelles méthodes de communications et de suivi de leurs requêtes, avec lesquelles le personnel de PIXEL ne s'était pas familiarisé. Ce changement a exigé une formation des employés de PIXEL, surtout ceux qui communiquent avec les clients, à propos de l'éthique des communications et des correspondances.

b. Le comportement des clients vétérans

Les anciens clients étaient habitués à contacter les développeurs, directement et pour n'importe quelle question ou aide. Chacun des clients était habitué à l'un des développeurs. C'est pourquoi, du fait que leur développeur « intime » n'est plus disponible tout le temps à les servir, cela a créé du mécontentement chez la plupart des anciens clients, de quoi exiger une rénovation des anciennes relations clients afin d'éviter les sentiments de mécontentement.

c. La surcharge des tâches

L'accroissement du nombre des clients a accru le nombre des tâches de développement et les requêtes de service. Or, avec une équipe relativement petite, les développeurs ont été totalement surchargés, ce qui les a forcés à travailler durant les weekends et les vacances officielles. Cette situation n'était qu'une solution temporaire qui a affecté à la longue la qualité des travaux et a obligé l'entreprise d'accroître son équipe de travail.

d. La mise à jour des différentes versions de logiciels

L'organisation des différentes versions des logiciels chez les clients a été réalisée d'une manière primitive grâce à quelques documentations simples. Cette manière s'est avérée efficace pour un nombre restreint de clients, mais très compliquée avec le nouveau nombre de client, et ne fournissant que le minimum de données nécessaires. Cela a conduit à trouver une nouvelle façon, systématique et dynamique, d'organiser les différentes versions, afin de lutter contre les défaillances et d'arranger la mise à jour des logiciels.

e. Le centre d'appels et le réseau informatique vieilliss

Le centre d'appels et de communications de l'entreprise qui comporte deux lignes téléphoniques et des appareillages classiques, est devenu chétif, devant le nombre de communications nécessaires pour assister tous les clients.

De même, le réseau informatique de l'entreprise n'était plus apte ni à comprendre le nombre de stations nécessaires ni à gérer les accès distants, qui sont devenus nécessaires pour servir les clients à l'étranger, durant les weekends et les vacances au Liban qui parfois, sont des jours de travail dans leurs pays.

Cette situation a sollicité une solution catégorique du réseau informatique et des équipements de communications.

Tous ces changements, en plus de certains autres, ont exigé un changement du comportement : un comportement de professionnalisme, basé sur des objectifs clairs et bien structurés.

3. Dégagement des objectifs

Les besoins exigés par les changements récents ont conduit l'administration à analyser la nouvelle situation de l'entreprise, à fixer des objectifs nouveaux, à courts et moyens termes, qui devraient être achevés en parallèle avec les objectifs majeurs de l'entreprise.

Les objectifs dégagés ont été regroupés en trois catégories : Organisationnels, Matériels et Logiciels. Pour chacune de ces catégories, un ordre de priorités a été réalisé, selon leurs exigences et leur faisabilité.

Le tableau suivant (Table I) liste les différents objectifs dégagés et leurs priorités associées.

Table I : Les objectifs dégagés par l'administration de l'entreprise.

Priorité	Objectif
Catégorie no. 1 : Organisationnels	
1	Réorganiser l'équipe de travail
2	Structurer une hiérarchie pour le service client
3	Recruter, et renforcer le personnel
4	Intégrer une culture de communication professionnelle
Catégorie no. 2 : Matériels	
1	Renforcer le réseau informatique
2	Implémenter un centre d'appels moderne
Catégorie no. 3 : Logiciels	
1	Automatiser le moyen de suivi des problèmes
2	Automatiser le service client
3	Automatiser la mise à jour des versions des différents logiciels

Un plan d'action est rédigé par l'administration de l'entreprise ; il comporte toutes les démarches et les activités nécessaires pour atteindre les objectifs fixés dans un temps minimal.

Ce plan est le résultat des différentes solutions proposées par les membres de l'équipe de PIXEL. Il a commencé au début de l'an 2009 par une restructuration complète de l'entreprise, surtout du département du service client, et par une enquête de recrutement pour renforcer le personnel de l'entreprise. Encore faut-il souligner que, le plan a prévu un budget pour améliorer le réseau informatique et le centre d'appels de l'entreprise.

De même, le plan d'action a englobé la décision d'acquérir un ou plusieurs logiciels qui couvriront les trois objectifs de la catégorie no. 3.

C'est à la lumière de plusieurs recherches (détaillées dans les chapitres suivants), discussions et débats entre les membres les plus âgés de l'équipe, qu'a été prise pour deux raisons majeurs la décision de développer le logiciel recommandé par l'équipe de développement de l'entreprise. La première, d'ordre technique, est la crainte de ne pas pouvoir atteindre la solution souhaitée et toutes les mises à jour nécessaires dans une courte période. La deuxième, d'ordre administratif. En effet, l'administration a considérée déraisonnable l'achat

d'un logiciel, tout en possédant une équipe de développement et pas mal de ressources (bibliothèque, composants graphiques etc.) capables de faciliter le développement du logiciel.

Ainsi, tel a été le début de la conception du projet, sujet de ce mémoire.

Ce projet est planifié pour englober les trois objectifs de la catégorie no. 3, mais au fur et à mesure, selon leurs priorités, depuis l'organisation de la gestion et du suivi des problèmes signalés par les clients, concernant les défaillances, les problèmes et les erreurs de leurs logiciels, jusqu'à la gestion des modifications des différentes versions des logiciels, en passant par la gestion de toutes les données nécessaires aux clients.

Le projet nécessitant un temps large de développement, ce qui n'est pas réalisable vu les surcharges et le manque du personnel, l'administration a fixé les objectifs prioritaires suivants, qui devraient être couverts par sa première version :

- a- Organiser les coordonnées des clients de l'entreprise.
- b- Gérer le suivi des problèmes reportés par les clients.
- c- Couvrir le maximum des canaux de communications.
- d- Procurer à l'entreprise une base de données qui offre à l'administration des résultats statistiques des différentes activités d'assistance et de service.

Une équipe de quatre personnes est composée pour développer cette première version du projet. Etant membre de cette équipe et ayant décidé, de devenir le responsable du département du service client, j'ai été nommé chef de ce projet.

Une période initiale de six mois est fixée pour achever le développement de cette première version. Cette période étant relativement insuffisante par rapport aux fonctions demandées, l'assistance de tout le personnel a été fortement recommandée, surtout durant les phases de tests des différentes fonctions du projet.

Deuxième partie

Etude bibliographique

Cette partie décrit les résultats des recherches réalisées pour mieux comprendre toutes les composantes du sujet du projet. Elle met en relief l'importance de la gestion de la relation client et sa portée par rapport au marketing et à l'évolution des entreprises. Puis, elle propose une définition du service client à multi-tiers, et précise ses avantages et ses inconvénients. Enfin, elle donne une explication des bugs informatiques, leurs classifications, leurs cycles de vie et le dégât qu'ils pourront produire.

A la fin de cette partie, un résumé des différents logiciels disponible actuellement sur le marché et qui traitent les fonctions de suivi de problèmes et de service client, leurs principales fonctionnalités.

Chapitre III : Gestion de la relation client

Le client est généralement la principale source de revenus des entreprises. Avec le changement de l'économie, notamment dû à l'intégration des nouvelles technologies dans les relations client-entreprise, la concurrence devient de plus en plus serrée et les clients peuvent ainsi se permettre de choisir leur fournisseur ou de le changer par un simple clic.

Dans ce chapitre sera définie la notion de gestion de la relation client, sa première apparition dans le domaine du marketing, et ses avantages pour l'évolution des entreprises. De même, ce chapitre présente les différents canaux de communications avec le client, surtout les plus utilisés de nos jours, qui devraient être couverts par tous logiciels gérant la relation client.

1. Définition

La gestion de la relation client (GRC), en anglais Customer Relationship Management (CRM), vise à proposer des démarches et des méthodes technologiques pour renforcer la relation client-entreprise, séduire des nouveaux clients et conserver les clients vétérans. La GRC n'est pas un outil ; c'est une démarche qui engage l'entreprise à mettre les clients en tête de ses priorités. Elle représente aux entreprises un avantage et un facteur de réussite.

Les fondamentales du GRC se résument par les quatre points suivants :

a- Connaissance du client

Plusieurs informations sont nécessaires pour développer des relations à long terme comme l'historique des achats, le mode de paiement préféré, les moyens de communications etc.

b- Stratégie relationnelle

Les entreprises qui mesurent leurs réussites au nombre des transactions réalisées et par leur chiffre d'affaire, ne manifestent qu'un intérêt limité envers le client. Tandis que celles qui développent une stratégie relationnelle s'intéressent au développement des relations à long terme en débutant par les clients potentiels.

Cette relation commence par la transaction commerciale et s'approfondit sur la base de confiance et d'engagement mutuels et croissants.

c- Communications

Une stratégie relationnelle devrait avoir la capacité à gérer un dialogue avec les clients au moyen des réseaux de communications intégrés, permettant de communiquer partout et à tout moment.

d- Proposition de valeurs personnalisées

Un dialogue véritable avec le client doit déboucher sur la création des propositions personnalisées tant au niveau de l'offre du produit que de celui du tarif et du service.

Il existe deux visions de la GRC :

- Technique, qui met l'accent sur la collecte et le partage des informations, la personnalisation des communications.
- Stratégique, qui met l'accent sur la valorisation du capital client et de la personnalisation de l'offre.

2. Marketing transactionnel et Marketing relationnel

Le marketing transactionnel ou « marketing classique », est l'ensemble des techniques de marketing focalisées sur l'acte de vente. Il englobe les techniques de ventes entendues au sens large et le marketing du produit. Tandis que le marketing relationnel est une politique et un ensemble d'outils destinés à établir des relations individualisées et interactives avec les clients, en vue de créer et d'entretenir chez-eux des attitudes positives et durables à l'égard de l'entreprise ou de la marque.

En effet, R.P Bagozzi en 1975 a énoncé la première fois le concept de marketing comme étant un processus renouvelé d'échanges entre acheteurs et vendeurs, ce qui a posé implicitement la problématique de l'évolution de la relation client (2), quitte à ce que, les domaines du marketing des services seront apportés par la suite au marketing relationnel.

L'école scandinave du marketing est la première à établir des comparaisons précises entre ces deux concepts de marketing (3). Cette comparaison pourrait être résumée par le tableau ci-dessous (Table II).

Table II : Comparaison entre marketing Transactionnel et Relationnel.

	Marketing Transactionnel	Marketing Relationnel
Perspective temporelle	Court terme (approche « one shot »)	Long terme (approche continue)
Approche marketing dominante	Marketing mix	Marketing interactif (soutenu par le marketing mix)
Composante stratégique principale de l'entreprise	Dimension objective (approche produit)	Dimension relationnelle (solutions proposées)
Mesure de la satisfaction du client	Contrôle de la part de marché (approche indirecte)	Gestion des bases de données relationnelles (approche directe)
Système d'information sur les clients	Enquêtes de satisfaction (mesures épisodiques)	Systèmes de feedback en temps réel (mesures instantanées)
Interdépendance entre fonctions de l'entreprise	Cloisonnement entre fonctions et interfaces limitées (organisation verticale et hiérarchique)	Approche transversale et importance des interfaces (organisation horizontale et collaborative)

Le concept de marketing relationnel se concentre sur les interactions entreprise-client afin de mesurer la satisfaction des clients et leurs feedbacks en temps réel, ce qui met en lumière l'enjeu de GRC et son importance de point de vue du marketing contemporain.

3. Multi-canal de communication

L'intégration de nouvelles technologies, surtout ceux de communications, dans la relation client-entreprise, a offert aux clients une diversité de canaux afin de contacter leurs entreprises. Cette vaste gamme pousse chaque entreprise intéressée au GRC à couvrir le plus possible de ces canaux.

Le multi-canal est l'intégration de tous canaux de communications dans un même système capable de gérer les interactions avec les clients. On distingue les canaux "One to one" (email, courrier, téléphone, web Call Back) et "One to many" (site web, emailing, catalogue papier, publicité).

MARKESS International, le cabinet d'études et de conseil spécialisé dans l'analyse de la modernisation et de la transformation des organisations privées et publiques avec les technologies de l'information, a menée une étude en 2009 intitulée « Optimisation de la gestion des interactions clients avec internet et le mobile » (4). Un des objectifs de cette étude était de savoir quel est le canal le plus favorisé aux clients.

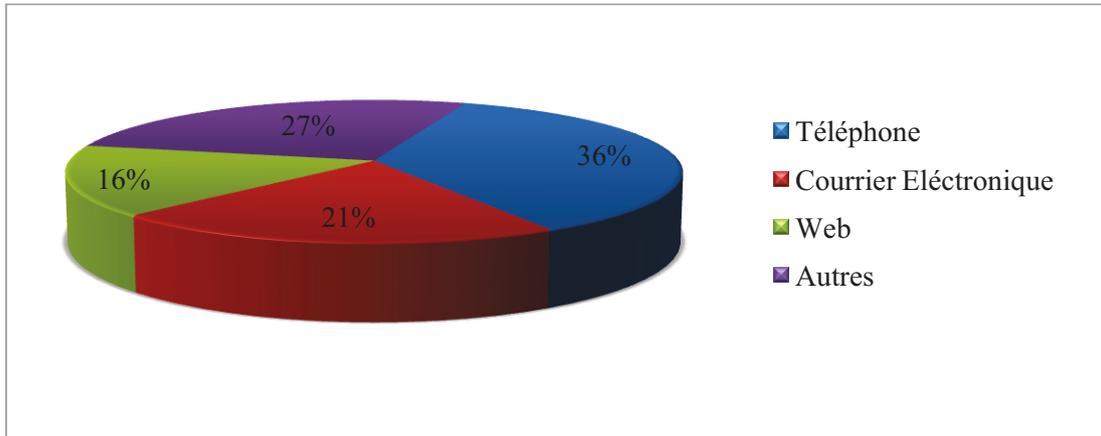


Figure 3 : Résultats de l'étude « Optimisation de la gestion des interactions clients avec internet et le mobile » réalisée par MARKESS International.

Les résultats de cette étude (affichés par la Figure 3) montrent que parmi tous les canaux de communication, le téléphone reste le canal privilégié des interactions avec les clients, pour 36% des interactions.

Dans cette même étude, les perspectives à l'horizon 2010 montrent que le téléphone devrait marquer un recul jusqu'à 28% des interactions au profit du courrier électronique et du web.

Enfin, actuelle diversité des canaux de communications et les résultats de plusieurs études, font du multi-canal une exigence à toute gestion de relation client survenant et à tous les logiciels GRC.

4. Avantage du GRC

Une mauvaise gestion de la relation client peut avoir plusieurs conséquences graves sur les entreprises. Ces conséquences sont résumées par des statistiques regroupées par Thierry SPENSER, après sa lecture du livre « Marketing Genius » de Peter FISK (5) :

- 20% de vos clients font 80% de votre chiffre d'affaires
- 10% de vos clients font 90% de vos profits
- Un client très satisfait en parle à 3 personnes
- Un client mécontent en parle à 12
- Un client très mécontent en parle à 20
- 98% des clients mécontents ne se plaignent jamais, ils se contentent de vous quitter
- Les expériences négatives pèsent pour 65% des clients perdus

- 75% des expériences négatives n'ont pas de rapport avec le produit
- La raison principale pour laquelle les clients vous quittent est le fait qu'ils ne se sentent pas vraiment respectés.
- Recruter un client coûte trois fois plus cher que de le fidéliser.
- Reconquérir un client mécontent coûte 12 fois plus cher.
- Sur une période de 5 ans, une société arrive à conserver en moyenne 20% de ses clients.
- 5% d'augmentation de la fidélité augmente vos profits de 25 à 55%

Enfin, la GRC apporte aux PME de nouvelles façons de communiquer, d'exploiter et d'élaborer des stratégies pour gérer leurs personnels, leurs clients et leurs prospects. Ses avantages peuvent se résumer comme suit (6):

- Avantages économiques :
 - a- La fidélisation du client vient en premier.
 - b- L'amélioration de la qualité du service client.
 - c- L'augmentation de la productivité.
 - d- Le gain en parts de marché et un meilleur ciblage de sa clientèle.
- Avantages sociaux :
 - a- L'implication autour du client.
 - b- La responsabilisation.
 - c- La valorisation des compétences.

Chapitre IV : Service client

Le projet, thème de ce mémoire, étant destiné à organiser le service après vente au sein de PIXEL, il est important de définir le dit service de point de vue professionnel : son importance et les conséquences d'un mauvais service. En outre, au cours de ce chapitre, seront introduits le service à multi-tiers et ses avantages organisationnels.

1. Définition

Le service client ou service à la clientèle, un des composants destiné à être automatisé par la GRC, constitue la prestation des services des clients avant, pendant, et après l'achat.

Selon JAMIER L. SCOTT (2002), « Le service client est une série d'activités qui visent à améliorer le niveau de satisfaction du client – c'est l'intention qu'un produit ou service à rencontrer les attentes du client » (Traduit de l'anglais) (7).

De même THE CHARTERED INSTITUTE OF MARKETING le définit par « Le service client est ce que fournit ton entreprise pour gagner la satisfaction du client » (8).

Le service à la clientèle permet aux entreprises de fidéliser leurs clients en se distinguant avantageusement les unes des autres. L'accroissement du volet « service » que proposent les entreprises les oblige à considérer d'autres variables dans les offres qu'elles font au marché, notamment représentées par les personnes, les processus et les éléments physiques rattachés aux services.

Une étude réalisée en Juillet 2006 par ACCENTURE (Conseil en management, services technologiques et externalisation), montre que 72% des consommateurs américains estiment que la qualité de service a une influence sur leurs décisions finales d'acheter ou non un produit (9).

D'après Don PEPPERS et Martha ROGERS (10):

« Les clients ont des souvenirs. Ils se souviendront de vous, si vous en souvenez ou pas »

« La confiance des clients peut être détruit soit par un problème de service majeur, soit par plusieurs petites démonstrations d'incompétence » (Traduit de l'anglais)

Aujourd'hui, la tendance est au développement du « Self-service », c'est-à-dire permettre aux clients de s'entraider à l'aide des forums en ligne par exemple. Mais le risque de cette dernière est de perdre le contact avantageux avec les clients. En cas de mécontente du client, il se peut qu'il trouve les réponses à ces interrogations dans d'autres communautés et forums.

2. « 5 P's » du service client

TICSI (The International Customer Service Institut) a élaboré un modèle nommé « The 5 P's of Customer Service » (11). Ce modèle préconise une approche flexible, encore mesurable, qui met en œuvre tous les éléments qui composent la prestation du service client. Il intègre et met l'accent sur l'importance des 5 P :

- La politique du service client (Policy)
- La qualité du produit / service (Product)
- L'environnement des clients (Premises)
- Le processus du GRC (Processes)
- Le personnel (People)

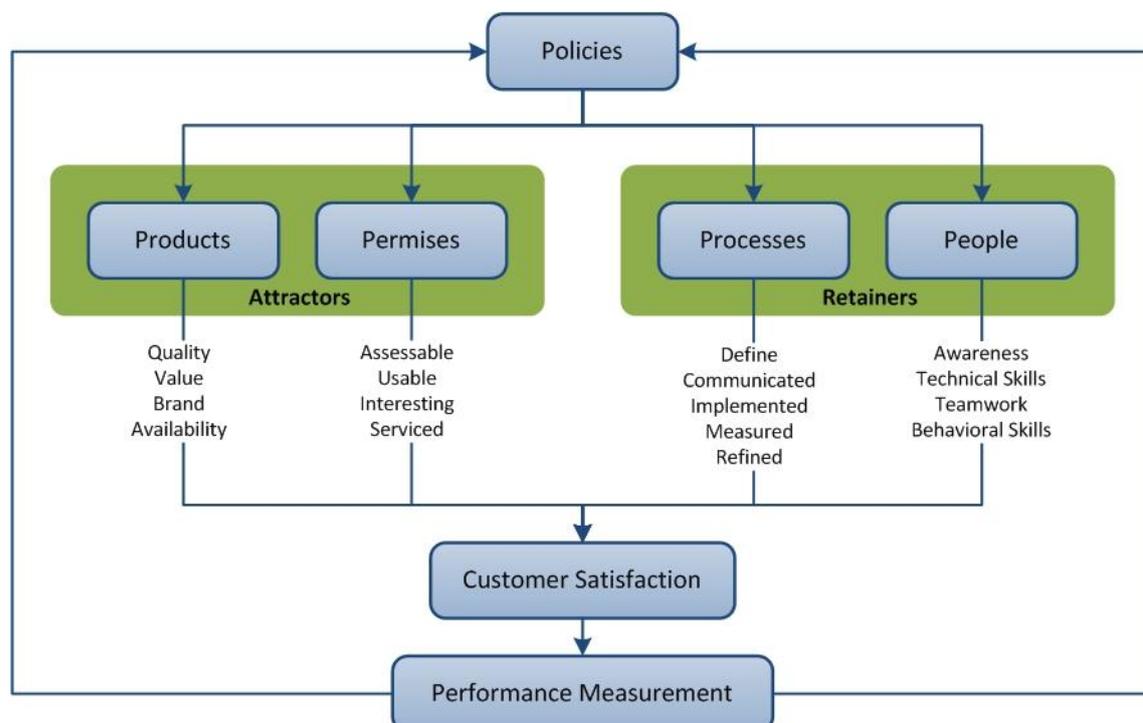


Figure 4 : Le modèle « 5 P's of Customer Service » de TICSI.

Comme pour toutes les opérations de l'organisation, ces composants de base ne déploieront pleinement leurs potentiels et leur valeur que s'ils sont intégrés dans l'infrastructure

organisationnelle et maintenue grâce à un cycle continu de mesure et de communication, soutenu par le pilier essentiel de l'éducation et de formation.

3. Service à Multi-tiers

Toutes les entreprises visent à offrir à leurs clients des services de haut niveau (HA – High Availability). Ces services HA incluent la fiabilité, la disponibilité et la serviabilité. Ils sont certainement liés aux technologies utilisées, mais d'avantage plus liés à l'organisation structurelle du processus de service (12).

Le modèle de service à multi-tiers est un modèle structurel, efficace pour offrir des services HA. Il est notamment utilisé par les entreprises des services informatiques. Dans ce modèle, les fonctions de l'entreprise sont placées dans une hiérarchie d'escalade avec des lignes de démarcation claires entre les différentes fonctions, pour s'assurer que les responsabilités et les devoirs sont clairement définis.

Le nombre de tiers ou niveaux dans un tel modèle de service diffère d'une entreprise à une autre, selon les besoins et les ressources de chacune.

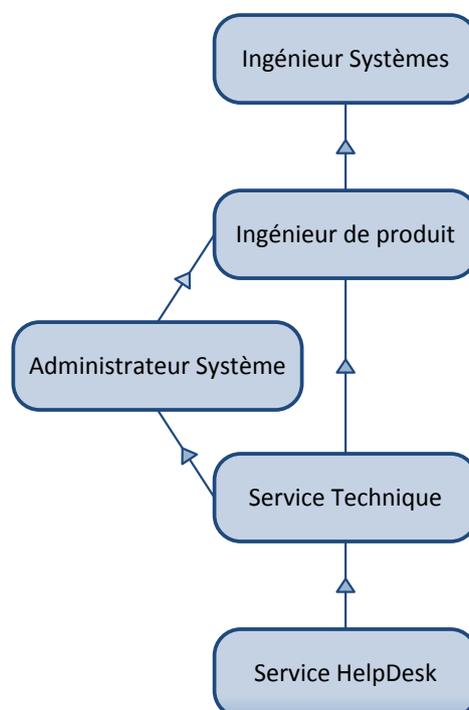


Figure 5 : Modèle de service Multi-tiers à 5 niveaux.

La Figure 5 illustre un exemple de modèle de service informatique à multi-tiers qui comporte 5 niveaux hiérarchiques, reliés par des connexions unidirectionnelles qui symbolisent les fonctions d'escalade entre les différents niveaux. Dans ce qui suit, une description des fonctions des différents niveaux de ce modèle et des transitions d'escalades d'un niveau à un autre.

- Niveau 1 : Service Helpdesk

Ce niveau est le premier en ligne de service. Il reçoit les appels des clients. Son personnel effectue le triage, la solution des problèmes simples eux-mêmes (éventuellement en utilisant des outils de gestion des connaissances) et fait passer le reste sur le niveau 2. Les membres de ce groupe sont peu qualifiés et techniquement peu formés. Ce niveau de service est souvent sous-traité, car les compétences nécessaires pour accomplir efficacement le travail ne sont pas aussi directement liées aux produits et services offerts, comme c'est le cas pour les autres niveaux.

- Niveau 2 : Service Technique

Ce groupe gère les escalades du niveau 1 et escalade au niveau 3 ou 4. Ses membres essaient de trouver la solution aux problèmes, soit par leurs propres moyens soit en utilisant des ressources à leurs dispositions. Ils ont généralement un contact direct aux niveaux 3, 4 et 5 et possèdent une compréhension technique des produits. Le membre du service technique se déplace souvent aux niveaux 3, 4 et 5 en fonction du développement de ses compétences.

- Niveau 3 : Administrateur Système

Les administrateurs système du niveau 3 poignent les escalades du niveau 2 et escaladent leurs requêtes au niveau 4. Le personnel de ce niveau 3 se concentre sur l'infrastructure sur laquelle s'appuient les services à haut niveau de fiabilité pour fonctionner correctement.

- Niveau 4 : Ingénieur de produit

Les ingénieurs de produit gèrent les escalades des niveaux 2 et 3, et escaladent au niveau 5. Ce niveau est formé d'ingénieurs techniquement qualifiés, qui se concentrent sur le fonctionnement du produit. Habituellement, leur spécialité fonctionnelle est l'ingénierie des systèmes, ils sont probablement des employés qualifiés qui ont travaillé pendant un certain temps au niveau 3.

- Niveau 5 : Ingénieur Système

Ces ingénieurs sont chargés de développer et maintenir les produits. Ils reçoivent les escalades du niveau 4. Ils sont responsables du suivi des défauts du produit et de surveiller l'état du produit qu'ils ont développé et qu'ils sont au cours de maintenir.

L'utilisation d'un modèle de service à multi-tiers au lieu d'un seul groupe de service, assure le meilleur service de la manière la plus efficace et minimise la perte de temps des personnes les plus compétentes de l'équipe. Mais la réussite d'un tel modèle dépend de la formation du personnel ; la compréhension des fonctions et des responsabilités et le moment où un problème devrait être escaladé et à quel niveau.

4. Le coût d'un mauvais service

Une question fréquemment posée :

« *Le service client, très coûteux à réaliser, ou nuisible une fois négligé?* »

En effet, dans les petites et moyennes entreprises, 80% des activités de service client n'ajoutent pas ou ajoutent peu de valeur à l'entreprise en termes d'augmentation des revenus ou de protection de la marge. Pourtant celles qui réussissent parmi ces dernières, y parviennent grâce à l'amélioration et à l'importance accordée à ces activités (13).

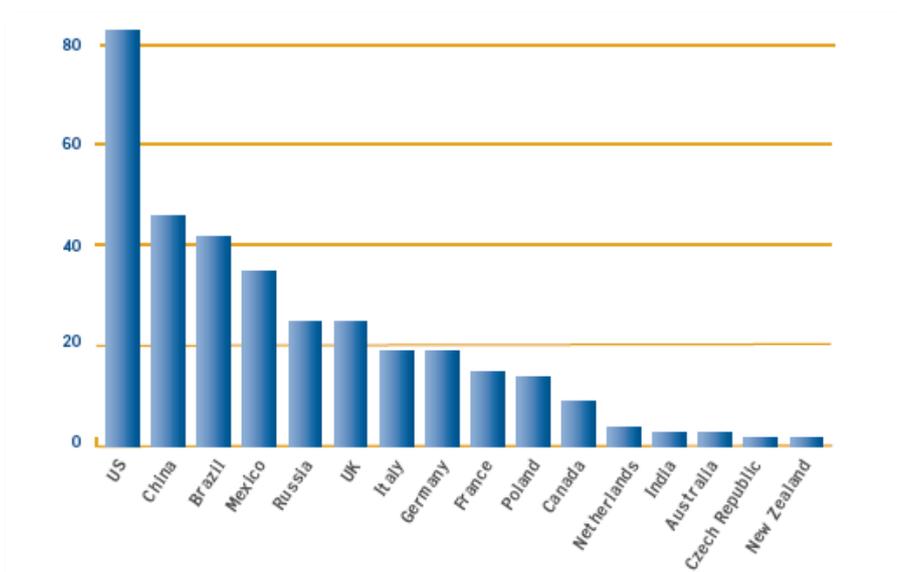


Figure 6 : Résultats de l'étude « The Cost of poor customer service : the economic impact of the customer experience and engagement.

Une étude à grande échelle commandée par GENESYS, conduite par Greenfield Online, sur les impacts d'un mauvais service client est publiée en Décembre 2009. Réalisée dans 16 pays industriels, elle est intitulée « The Cost of poor customer service : the economic impact of the customer experience and engagement » (14).

Cette étude a évalué une perte de 338.5 milliards de dollars par an (voir les résultats affichés dans la Figure 6). Cette perte est estimée au nombre des contrats résiliés et des ventes annulées. La perte est définie par le fait que 63% des transactions étaient attirées par un concurrent et 37% totalement abandonnées.

Les 63% des pertes dont profitent les concurrents, montrent que les conséquences d'un mauvais service qui engendre une insatisfaction des clients, peut coûter à l'entreprise une perte énorme inattendue et parfois irrécupérable.

Chapitre V : Systèmes de Suivi et d'Assistance

Afin d'offrir des services HA (High Availability) aux clients, les entreprises utilisent des logiciels spécialisés pour organiser les différentes requêtes et données. Ces logiciels catégorisés en systèmes de suivi des problèmes et en systèmes de gestion de service d'assistance, sont fréquemment utilisés de nos jours, notamment dans les entreprises qui offrent des services informatiques.

Dans ce chapitre, afin de pouvoir développer un tel logiciel, les principaux objectifs des systèmes de suivi et ceux d'assistance seront définis, la différence entre eux et le principe de ticket d'incident utilisé par la plupart de ces systèmes seront aussi présentés. En plus, les différentes catégories des bugs informatiques seront exposés, ainsi que leur classification et leur cycle de vie.

1. Définition

Les systèmes de suivi des problèmes et les systèmes de gestion du service d'assistance semblent identiques à la première impression. Cependant, ils se différencient par leurs objectifs principaux. Pourtant, ils pourront être unis dans un même système qui regroupe tous leurs objectifs.

1.1 Système de suivi des problèmes

Les systèmes de suivi des bugs (ou des problèmes) sont connus en anglais sous plusieurs noms : Issue Tracking Systems, Trouble Ticket Systems, Support/Incident Ticket Systems, Bug Tracking System etc.

Notamment utilisés dans les entreprises qui développent les logiciels, ces systèmes aident les entreprises à améliorer la qualité de leurs produits. Ils sont utilisés durant deux stades différents du cycle de vie des produits :

- La phase de développement

Dans ce stade, ces systèmes sont utilisés par le personnel de contrôle de qualité (QA). Ils servent à gérer les différentes erreurs ou dysfonctionnements des produits afin de suivre les problèmes communiqués aux développeurs.

- La phase de maintenance

A ce stade, ces systemes sont utilises afin d'ameliorer le service d'assistance client. Ils permettent de suivre la progression des bugs reportes par les clients tout au long de leurs traitements.

Ces systemes permettent aux utilisateurs (personnel de QA ou client) de reporter directement les problemes rencontres (techniques, fonctionnels, demandes d'amelioration etc.), de sorte que les developpeurs mis au courant de tous ces problemes, effectuent les traitements necessaires.

La base de donnees est le composant principal de ces systemes. Elle enregistre toutes les caracteristiques des bugs : la problematique, la severite, l'identite de la personne qui l'a reporte, et la personne qui doit le resoudre.

Ces logiciels incluent le concept de « Cycle de vie d'un bug » qui commence par la date ou le bug a ete reporte tout au long de son traitement jusqu'a sa solution.

1.2 Systeme de gestion du service d'assistance

Les systemes de gestion des services d'assistance sont des logiciels conus pour gerer toutes les fonctions du service client.

En principe, ces logiciels doivent couvrir les processus d'ITIL (Information Technology Infrastructure Library) v2 *Support service*. En effet, ITIL est un ensemble d'ouvrages recensant les bonnes pratiques pour la gestion de services informatiques, edictees par l'Office public britannique du Commerce. La deuxieme version de cette bibliotheque comporte 6 livres regroupes sous le nom *Support Service* et qui decrivent comment s'assurer que le client a acces aux services informatiques appropries. Les 6 livres d'ITIL v2 *Support Service*, sont les suivants :

1. Le centre de service (Service Desk)
2. La gestion des incidents (Incident Management)
3. La gestion des problemes (Problems Management)
4. La gestion des changements (Change Management)
5. La gestion des mises en production (Release Management)
6. La gestion des configurations (Configuration Management)

C'est évident qu'ITIL a distingué entre « Incident » et « Problèmes ». En effet plusieurs incidents peuvent être liés au même problème noté.

En pratique, les logiciels de gestion des services d'assistance qui ne couvrent pas tous les processus d'ITIL, surtout ceux qui ne distinguent pas entre problème et incident, sont considérés comme des logiciels de suivi des problèmes.

2. Principe des Tickets d'incident

La plupart des logiciels de suivi des problèmes utilisent le concept des tickets d'incident (ou de suivi). Ce principe consiste à sauvegarder toutes les requêtes reportées par les utilisateurs en forme de dispositif de ticket. Ce dispositif permet d'indiquer l'évolution du traitement du défaut indiqué, la date probable de résolution etc.

Généralement, ces tickets sont affectés par un indicateur de priorité. Cet indicateur est géré par les administrateurs ou les logiciels eux-mêmes, selon la méthode de gestion des priorités internes de l'entreprise qui, obligatoirement, doit avoir comme facteur principale la sévérité du problème.

En fait, quand le concept des tickets d'incidents a commencé, il a été évalué aux tableaux accrochés tout près des lits des patients dans les hôpitaux. Ces tableaux qui débutent par le problème, se remplissent au fur et à mesure par le progrès, les travaux réalisés durant les différents stades et les différentes personnes (15).

Bien que ce principe émerge dans les centres d'appels, sa fonction principale reste la même : suivre le problème dans les phases de détection, communication et de résolutions.

De nos jours, ce concept est largement utilisé dans le développement des applications de suivi et d'assistance, surtout dans celles basées web ; un client qui a un problème peut créer plus facilement un ticket que remplir un formulaire ou attendre sur les lignes de téléphones.

3. Gestion de priorité des bugs

Dans le monde informatique, le terme Bug est utilisé pour désigner tout type de dysfonctionnement des programmes, qu'il soit mineur (erreur d'affichage) ou majeur (crash

de système), ce qui engendre le besoin de classification des bugs selon leurs conséquences ou leurs sévérités, pour ainsi pouvoir définir leurs priorités de traitement.

En général, il n'existe pas une classification reconnue des bugs. Plusieurs classifications ont été réalisées, toutes aboutissant au même résultat : La gestion des priorités des bugs (implicitement des tickets qui les représentent). Cette gestion sert à déterminer l'importance de fixer le bug par rapport aux autres. Les classifications les plus utilisées et les plus logiques sont celles basées sur la sévérité du bug ou le dégât produit par le bug.

Dans ce qui suit un exemple de classification de bugs selon leurs sévérités (16), associé à des exemples d'un système de gestion financière pour mieux comprendre les conséquences de chacune.

Table III : Classification des Bugs par sévérité.

Sévérité	Conséquences
Bloqueur	L'application ou un des ses composants crash ou ne démarre pas. Déchéance de données.
	<i>Manque d'un fichier composant du système (DLL), ce qui empêche le lancement du système.</i>
	<i>Les numéros de séries générés par les déclencheurs de la base de données ne sont pas correctes (Triggers).</i>
Critique	Une fonction principale ne fonctionne pas, ne peut pas être utilisés ou renvoie des résultats incorrects.
	<i>Les écritures comptables des transactions ne sont pas correctement produites par le système et nécessitent une intervention des utilisateurs.</i>
Majeur	Une fonction principale est difficile à utiliser. Une fonction secondaire ne fonctionne pas, ne peut pas être utilisée ou renvoie des résultats incorrects.
	<i>Les comptes des clients ne sont pas récupérés par le système après la sélection du client.</i>
	<i>En changeant la devise utilisée, les montants ne sont pas automatiquement convertis.</i>
Normale	Une caractéristique secondaire ne fonctionne pas, ne peut pas être utilisée ou renvoi des résultats incorrects. Une caractéristique mineure ne fonctionne pas, ne peut pas être utilisée ou renvoie des résultats incorrects.
	<i>Un rapport statistique n'affiche pas la devise des transactions.</i>
	<i>Les relevés des comptes ne pourront pas être sauvegardés en fichier Excel.</i>
Mineur	Une fonction secondaire a un problème d'affichage. Une fonction mineure est difficile à utiliser.
	<i>Les noms des clients ne sont pas imprimés en caractères capitales.</i>
	<i>La tabulation ne fonctionne pas correctement dans les interfaces graphiques.</i>

Sévérité	Conséquences
Trivial	Erreurs d'affichage mineur, fautes grammaticales non graves etc.
	<i>L'interface graphique affiche le libellé « Cleint » au lieu de « Client ».</i>
Réconfort	Amélioration des fonctions du produit, en raison d'une mauvaise utilisation ou en fonction de commentaires des utilisateurs. Ce n'est pas une exigence de conception du produit.
	<i>Récupérer automatiquement les données d'une transaction après insertion de son numéro de série, sans besoins d'utiliser l'interface de recherche.</i>

Pour chacune de ces classes, un niveau de priorité pourra être associé (16). Dans ce qui suit (Table IV) une association de priorités basée sur la sévérité du bug et son fréquence d'occurrence.

Table IV : Association de priorité par sévérité et fréquence du bug.

Sévérité	Occurrence	Priorité
Bloqueur	-	doivent être réparés
Critique	-	doivent être réparés
Majeur	Fréquente	doivent être réparés
	Rare	recommandés d'être réparés et doivent être planifiés
Normale	Fréquente	recommandés d'être réparés et doivent être planifiés
	Rare	agréables à traiter mais non planifiés
Mineur	Fréquente	agréables à traiter mais non planifiés
Trivial	Fréquente	agréables à traiter mais non planifiés
	Rare	pas susceptibles d'être réparés rapidement
Réconfort	-	pas susceptibles d'être réparés rapidement

En réalité, la sévérité et la fréquence d'occurrence du bug ne sont pas les seuls facteurs de la gestion des priorités des bugs. Parfois l'importance d'un client peut augmenter la priorité d'un bug *Mineur*, qui devient par le fait même prioritaire par rapport à un autre bug *Majeur*, sans oublier que d'autres facteurs pourront entrer en jeu. Tout cela dépend de la stratégie et du management interne des entreprises.

4. Cycle de vie d'un bug

Le cycle de vie d'un bug (Bug Life Cycle en anglais) est l'ensemble des états où pourrait se trouver le bug, implicitement le ticket qui le représente, et toutes les conditions des transitions du bug d'un état à un autre. Ce cycle de vie n'est défini par aucune norme ou standard ; il est spécifique pour chaque organisation et chaque logiciel.

Sous Bugzilla, un des logiciels de suivi des problèmes les plus connus, le cycle de vie est assez détaillé et regroupe la pluparts des états d'un bug (Figure 7) (17).

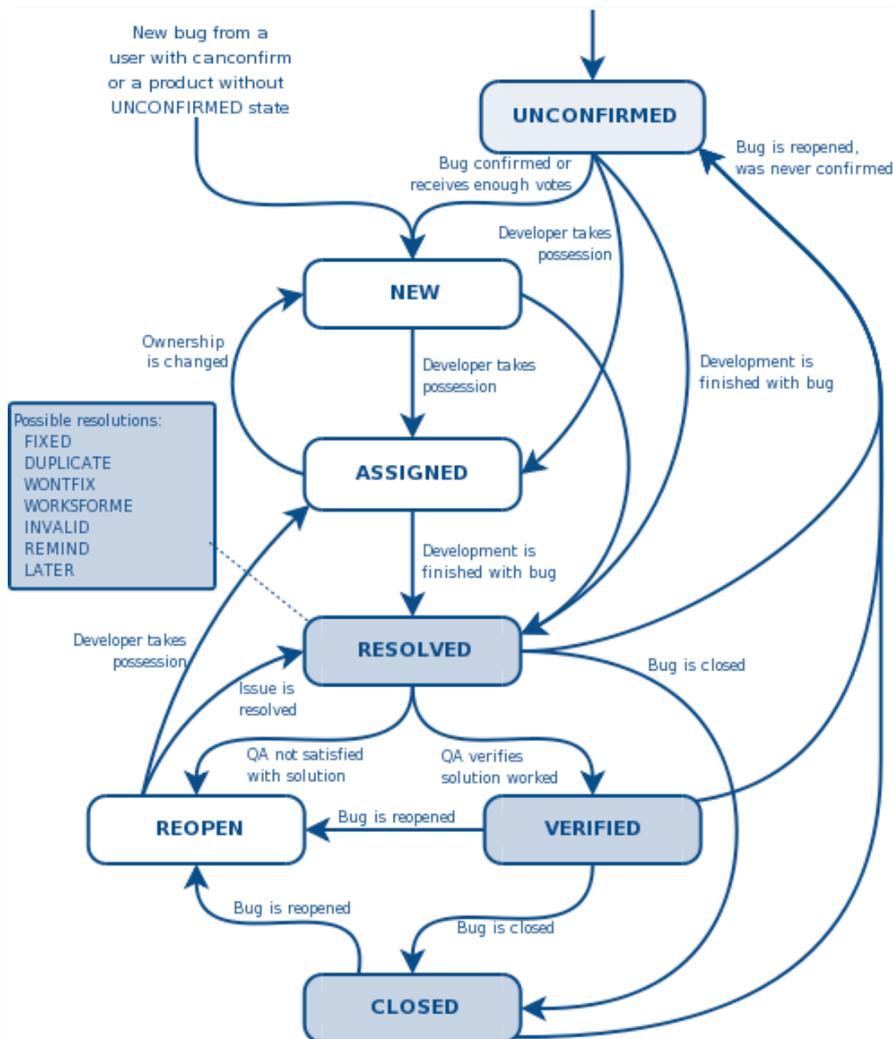


Figure 7 : Cycle de vie d'un bug selon Bugzilla.

Dans ce modèle de cycle de vie, un bug reporté doit être confirmé avant d'être confié à un développeur pour le traiter. Dès que le bug est confirmé, il serait assigné à un développeur qui commence son traitement. Une fois le traitement est terminé, le QA vérifie que le bug est

fixé et par la suite le bug sera fermé. Au cours du traitement, le bug pourrait être fermé pour plusieurs raisons comme la duplication. De même, un bug fermé peut être rouvert au cas où l'erreur / problème serait apparu de nouveau.

Un autre modèle de cycle de vie publié sur Buzzle.com (18), utilise les mêmes états du modèle précédent, mais avec 6 états supplémentaires (voir Figure 8) :

- *Deferred*: le bug sera différé s'il ne nécessite pas l'attention.
- *Postponed*: le traitement sera retardé au cas de disponibilité de données.
- *Pending Retest / Retested*: utilisés pour marquer que le bug, après le traitement, est mis en attente pour le test ou il est en cours de test.
- *Pending Reject / Rejected*: le bug sera marqué en attente de refus, ensuite refusé si le développeur ne parvient pas à régénérer le scénario dans lequel il était trouvé.

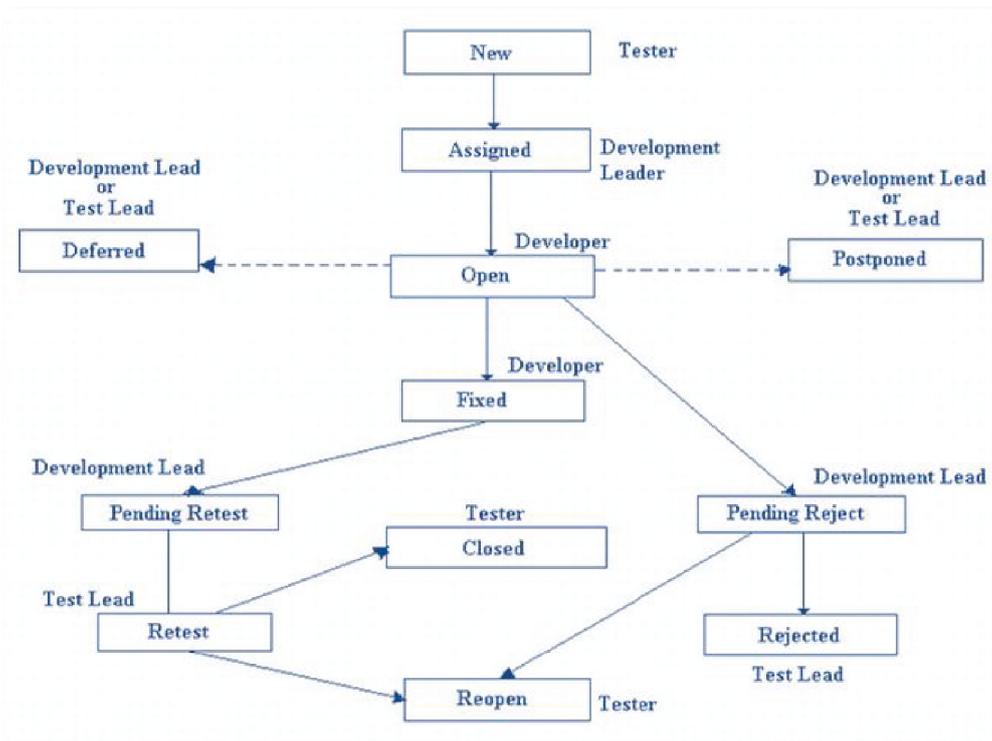


Figure 8 : Cycle de vie d'un bug selon Buzzle.com.

Ces deux cycles de vie pourront être adaptés dans la phase de test d'un logiciel et de même dans la phase de maintenance. Dans la première phase, le bug serait reporté par un membre de l'équipe QA (Quality Assurance) et dans la deuxième il est reporté par le client.

Chapitre VI : Solutions disponibles

Ce chapitre exposera les résultats des recherches effectuées sur les logiciels et les systèmes disponibles qui incorporent les fonctionnalités de service client souhaitées d'être développées par PIXEL pour couvrir ses besoins. De même, ce chapitre comporte des comparaisons de ces solutions trouvées concernant les technologies utilisées et les fonctions de gestion de la relation client intégrées, et implicitement celles du service client.

1. Enquête et critères

Les systèmes de suivi des problèmes et les systèmes de gestion des services d'assistance étant fréquemment utilisés de nos jours, les logiciels disponibles trouvés ont été donc très nombreux. Afin de limiter ces recherches, cinq systèmes ont seulement été sélectionnés. Ces cinq produits choisis sont les plus populaires sur le Web, ils sont utilisés par des grandes sociétés du monde informatique et non informatique.

- **Helpdesk Pilot** (19)

Développé par *TENMILES*, il est utilisé par IBM, D-Link, LUNZER+PARTNER, YOUSENDIT, Georgia State University, West Virginia University... il coûte vers les 1800\$ pour une édition d'entreprise comportant les composants basiques du système, 30 licences d'utilisation, 6 mois de service d'assistance et de mise à jour.

- **OMNITRACKER / E-Tracking** (20)

Développé par *OMNINET*, il est utilisé par BMW AG, Union IT-services, Rabobank International, Le Parlement Européen, Philips Technologie, SANYO Component Europe... Etant un système modulaire, il connaît un prix aléatoire, qui varie selon les modules et les interfaces choisis.

- **NetResults Tracker** (21)

Développé par *NetResults*, il est utilisé par Adobe Systems, Hewlett-Packard, Intel, XEROX, Bank of China, Deutsche Bank, Air Liquide, Princeton University... Ce système coûte 570\$ pour 10 licences d'utilisation, sans compter les fonctionnalités intéressantes comme la récupération automatisée des courriers électroniques, qui coûte 1195\$ pour chaque installation. De même, ce prix n'inclue pas le service d'assistance tarifé séparément.

- **Kayako Helpdesk SupportSuite (22)**

Développé par *Kayako Infotech*, il est utilisé par SEGA, Sun Microsystems, skype, FedEx, ACER, DHL, vodafone, HARVARD University, symantec... Pour chaque licence d'utilisation, ce système coûte 300\$, dont le frais de 6 mois de service d'assistance et de mise à jour.

- **Bugzilla (23)**

C'est le plus ancien et le plus connu des systèmes de suivi de problèmes, mis au service en 1998. Ce logiciel Open Source gratuit est utilisé par un grand nombre d'organisations et de projets comme Apache Project, Eclipse, NASA, Facebook, Yahoo, Nokia, The New York Times, RedHat, Novel et plusieurs autres.

Une décision de développement d'un logiciel, au lieu de l'acquisition d'un progiciel disponible au marché, a été déjà prise. C'est pourquoi les recherches se sont concentrées sur trois critères principaux:

- La catégorie du produit (Système de suivi des problèmes ou de gestion du service d'assistance).
- Le type des applications (basées Web ou Win).
- Les fonctions avantageuses intégrées dans chacun des systèmes trouvés afin d'améliorer la spécification du projet et ces fonctionnalités.

D'autres critères tels que les technologies de développement et le coût du produit, qui sont habituellement des critères principaux et parfois dominants dans ce type d'enquête, ont été exclus des recherches. En effet, ces critères n'ont été introduits dans les recherches que pour des raisons de curiosité professionnelle.

2. Etude comparative

Tous les systèmes sélectionnés utilisent le concept de ticket de suivi. Ils comportent des fonctions de manipulation et de gestion des tickets. De même, ils contiennent des modules de gestion des prospects. Ces logiciels diffèrent principalement par leurs catégories, les types des applications qu'ils offrent, leurs fonctions de management et les outils de statistiques qu'ils englobent. Ces différences seront résumées par les tableaux comparatifs qui suivent.

2.1 Catégories des systèmes

Les systèmes sélectionnés sont catégorisés en système de suivi de problèmes et système de gestion du service d'assistance (ou service client). Ces deux catégories se différencient par les fonctionnalités intégrées par chacune. Il est important de rappeler qu'un système de gestion du service d'assistance est implicitement un système de suivi des problèmes. Donc les systèmes de gestion du service d'assistance comportent toutes les fonctionnalités des systèmes de suivi des problèmes.

Certains systèmes de gestion du service d'assistance trouvés sont conformes avec ITIL et certifiés *Pink Elephant certified ITIL product*. Cela implique que ces systèmes distinguent entre la gestion des problèmes et la gestion des incidents, comme ils comportent des fonctions de gestion des changements et des configurations.

Le tableau suivant (Table V) résume les catégories des cinq systèmes sélectionnés.

Table V : Les catégories des systèmes disponibles trouvés.

	Suivi de problèmes	Gestion du service d'assistance	Notes
Helpdesk Pilot	X		
OMNITRACKER		X	Conforme avec ITIL v2 Support Service
NetResults Tracker		X	
Kayako Helpdesk	X		
Bugzilla	X		Open source

2.2 Types des applications

Les systèmes disponibles trouvés sont basés sur des applications bureautiques, ou des applications Web, et comportent parfois les deux types. En plus, certains systèmes comportaient des applications compatibles avec les assistants personnels numériques PDA (Personal Digital Assistant).

Le type des applications offertes par ces systèmes est résumé dans le tableau ci-dessous (Table VI).

Table VI : Les types des applications des systèmes disponibles trouvés.

	Application Bureautique	Application Web	Application PDA	Notes
Helpdesk Pilot		X		
OMNITRACKER	X	X	X	Les applications Web et PDA sont plutôt des interfaces conçues pour des fonctions limitées du système.
NetResults Tracker		X		
Kayako Helpdesk		X	X	
Bugzilla		X		

2.3 Canaux de communications

Tous les systèmes sélectionnés couvrent plusieurs canaux de communication afin d'offrir une meilleure gestion du service client, cela en gérant les relations client-entreprise par le canal de communication préféré par chaque client. On distingue les canaux suivants :

- Création directe des tickets par les employés de l'entreprise, pour les tickets reportés via appel téléphonique ou fax.
- Transformation automatisée des courriers électroniques récupérés en des tickets de suivi.
- Création indirecte des tickets par les clients via une application Web.
- Utilisation d'une base de connaissance (Knowledge Base), publiée sur le web, par les clients pour résoudre leurs problèmes.
- Lieu de discussion en ligne (Live Chat) qui permet aux clients de consulter les employés du service client.

Le tableau suivant (Table VII) résume tous les canaux de communication couverts par chacun de ces systèmes.

Table VII : Les canaux de communication couverts par les systèmes disponibles trouvés

	Téléphone Fax	Email	Interface client Web	Base de connaissance Web	Discussion en ligne	Notes
Helpdesk Pilot	X	X	X	X		
OMNITRACKER	X	X	X	X		Intègre une interface CIT (Computer Telephony Integration) qui facilite la récupération des données en recevant un appelle téléphonique.
NetResults Tracker	X	X		X		
Kayako Helpdesk	X	X		X	X	
Bugzilla	X	X				

2.4 Fonctions intégrées

Les fonctionnalités des cinq systèmes sélectionnés ont été exploitées d'après les documents de spécifications publiés par leurs développeurs et parfois par les versions démo valables. Les fonctionnalités essentielles traditionnelles de ces types de systèmes sont trouvées dans toutes leurs applications, mais les innovations ont différé d'un système à un autre.

En outre, tous les systèmes sélectionnés intègrent la fonction d'escalade des tickets de suivi d'un niveau de service à un autre, et couvrent implicitement l'architecture de service client à multi-tiers.

Toutes les fonctionnalités intégrées par les cinq systèmes choisis sont résumées dans le tableau suivant (Table VIII).

Table VIII : Fonctionnalités intégrées dans les systèmes disponibles trouvés.

Fonction / Caractéristique	Help Desk Pilot	OMNI-TRACKER	NetResults Tracker	Kayako Helpdesk	Bugzilla
Cycle de vie dynamique			X		
Calendrier des tâches				X	X
Détection automatisé de l'environnement de l'utilisateur (Système d'opérations, IP...)			X		
Gestion des contacts	X	X	X	X	
Gestion des tickets	X	X	X	X	X
Gestion des tâches et du travail d'équipe				X	
Gestions des emails	X	X	X	X	X
Interface Client	X		X	X	
Live Chat				X	
Multilingue	X	X			X
Notification interne				X	
Notification par email			X	X	X
Notification par SMS				X	
Porte CTI (Computer Telephony Integration)		X			
Rapports & Statistiques	X	X	X	X	X
Réponse prédéfinis (Canned Replies)	X				
Sécurités et autorisations	X	X	X	X	X
Self help ressources			X	X	
Suivi des activités (Activity Tracking)	X				
Supervision du staff (Staff Monitoring)	X				
Synchronisation avec d'autres logiciels		X		X	
Système de requête interne					X
Téléviser l'écran (Screen broadcast)				X	
Time Tracking	X	X	X	X	X

3. Portée des recherches

Après cette enquête, les objectifs des systèmes de suivi des problèmes et des systèmes de gestion du service d'assistance sont devenus plus clairs et lisibles. En effet, les résultats des recherches réalisées et surtout les fonctionnalités explorées des solutions disponibles ont facilité la description de la demande du projet.

PIXEL, comme c'est déjà mentionné, souhaitait développer un système de gestion complète de la relation client qui incorpore toutes les fonctions nécessaires pour offrir la meilleure assistance aux clients, ainsi que des fonctions de suivi des problèmes et des bugs. Mais comme un tel système nécessite un large temps de développement qui était hors de notre portée, la décision a été donc de développer un système qui sera par sa première version un système de suivi des problèmes et qui pourra intégrer dans ces futures versions des fonctions de gestion de la relation client afin de le transformer au fur et à mesure en une solution GRC. Cette première version a été planifiée sur le concept de ticket de suivi, en incorporant une architecture de service client à multi-tiers et toutes les fonctions nécessaires pour les escalades des tickets d'un niveau de service à un autre.

Les canaux de communication qui doivent être couverts par ce système ont été choisis selon la disponibilité des équipements à PIXEL, et selon la fréquence de leurs utilisations par les clients de l'entreprise.

La méthode de saisie directe des tickets qui couvre les requêtes reportées via téléphone ou fax a été exigée, parce que ces canaux sont fréquemment utilisés par les clients. En plus, la récupération automatique des courriers électroniques et toutes les fonctions nécessaires à la gestion de ces derniers, ont été fortement recommandées pour gérer tous les emails reçus par le service client.

Une base de connaissance publiée sur le web a été ambitionnée par l'entreprise. Mais à cause du faute de temps, la décision a été de permettre aux utilisateurs de saisir directement leurs requêtes via le web à l'aide d'une interface simple.

Les PDA étant inutiles pour le personnel de PIXEL, le développement d'une application compatible avec les assistants numériques n'a pas été une cible avantageuse. En effet, la conception a été concentrée sur deux applications : la première, bureautique, sera

utilisée par les employés pour toutes les fonctions de gestion des tickets, et la deuxième, application web, utilisée par les clients pour saisir directement leurs requêtes et poursuivre le traitement de leurs tickets déjà reportés.

Plusieurs fonctionnalités explorées ont été intégrées au projet, d'autres négligées. Le choix de ces fonctionnalités s'est limité aux objectifs du projet commandé par PIXEL, et aux ressources valables. Par exemple, l'intégration de la discussion en ligne (Live Chat) est fort important pour communiquer avec les clients et les servir instantanément, mais le développement de cette fonction aurait coûté trop du temps ce qui n'était pas disponible. De même, les réponses prédéfinies (Canned Replies) auront pu améliorer le service des clients en prédéfinissant des réponses aux questions fréquemment posées. Mais pour la même raison, cette fonctionnalité a été exclue. Toutes les fonctions intégrées dans le projet seront détaillées dans le premier chapitre de la partie suivante.

Enfin, la diversité des fonctionnalités intégrées dans un système de suivi de problème n'est pas le seul critère de son réussite. En effet, Bugzilla, parmi les cinq systèmes sélectionnés, intègre le moins des fonctionnalités, mais il est le système le plus ancien et est utilisé par un grand nombre de sociétés internationales.

Troisième partie

Solution proposée

Après la présentation de l'environnement du travail, les recherches réalisées et leurs portées, cette partie interprète les détails de la solution proposée et de sa réalisation. Elle comporte les différentes étapes de la mise en œuvre du projet, la description détaillée de ses composants et ses fonctions principales. De plus, les différents problèmes rencontrés ainsi que leurs conséquences sur le déroulement du travail seront décrits à la fin de cette partie.

Chapitre VII : Mise en œuvre

Ce chapitre traite les différents stages de développement du projet. Il comporte une description des objectifs du projet, des fonctions du produit et du choix des technologies utilisées. De même, ce chapitre interprétera la planification et l'organisation des différentes tâches de développement du projet, l'équipe du travail et la répartition des tâches sur les différents membres de l'équipe.

1. Description de la demande

Après l'analyse de sa situation actuelle (fin de l'an 2008) engendrée par l'accroissement fortuit et non organisé de son portefeuille, PIXEL a fixé les objectifs nécessaires à acquérir afin de résoudre ses problèmes rencontrés, spécialement le désordre produit par cet accroissement (voir Chapitre II : Problématique).

La solution désirée a été un système de gestion complète du service d'assistance et de gestion de la relation client conforme avec ITIL v2 *Support Service*. Mais le développement d'un tel logiciel vaste nécessite un temps de développement énorme, ce qui n'était pas valable à cette période. Ainsi, l'administration de l'entreprise a décidé de produire une première version de la solution souhaitée concrétisée par un système de suivi des problèmes (bug) qui couvre les besoins pressants et urgents de l'entreprise.

Ajoutons que l'administration de l'entreprise, était sûre et certaine que les clients mesurent les valeurs par les produits acquis et le service d'assistance après vente reçu, ce qui fait du service d'assistance un facteur principal de la réussite et de la continuité de l'entreprise. C'est pourquoi quelques fonctions additionnelles ont été recommandées par l'administration afin de soutenir la relation avec ces clients.

1.1 Objectifs

Comme toute solution ou projet développés, ce projet possède des objectifs clairs et spécifiques qui devraient être englobés par ses différentes fonctions. La plupart des objectifs du projet sont d'ordre classique pour tout système de suivi de problèmes, mais d'autres sont spécifiques, choisis par PIXEL pour couvrir ses besoins.

Les objectifs principaux du projet sont résumés par les 6 points suivants :

1. Organiser les coordonnées des clients de l'entreprise.
2. Accueillir les requêtes des clients.
3. Couvrir le maximum des canaux de communications.
4. Gérer le suivi des problèmes reportés par les clients.
5. Offrir aux clients la possibilité de suivre la progression de leurs requêtes.
6. Procurer à l'entreprise une base de données qui offre à l'administration des résultats statistiques des différentes activités de son service d'assistance.

Les objectifs du projet ont été traduits par différentes fonctions et activités intégrées dans le projet. Toutes ces fonctions seront détaillées par la suite.

1.2 Produit du projet

Le système est appelé B-WARN SUITE. Il est composé de deux applications ; une première bureautique, nommée B-WARN et une deuxième web, nommée LIVE B-WARN. La nomination B-WARN est une combinaison de la lettre « B », qui désigne le mot Bug, associée au mot « WARN » de l'anglais, qui signifie notifier ou avertir. Ces deux applications sont physiquement séparées mais connectées à une même base de données centralisées.

B-WARN est l'application principale du système qui effectue toutes les fonctionnalités attendues, alors que LIVE B-WARN n'est qu'une application complémentaire qui fournit des fonctions supplémentaires au projet.

Les utilisateurs du système, ou acteurs, sont regroupés en deux catégories:

- Les utilisateurs locaux, représentant les employés de l'entreprise, qui utilisent B-WARN et qui sont catégorisés selon les niveaux des services de l'entreprise.
- Les utilisateurs distants, représentant les clients de l'entreprise qui utilisent LIVE B-WARN.

Chaque utilisateur du système (local ou distant) est identifié par son nom d'utilisateur et son mot de passe. Cette authentification sert à initialiser les applications du système selon les accès et les autorisations offerts à l'utilisateur par l'administrateur du système.

Le système est basé sur l'idée des tickets. Toute erreur, demande ou question reportée par un des clients, est représentée dans le système par un ticket de suivi qui comprend toutes les données nécessaires à son traitement : date de création, nom du client, priorité, type, état, solution proposé, différents états de progression etc.

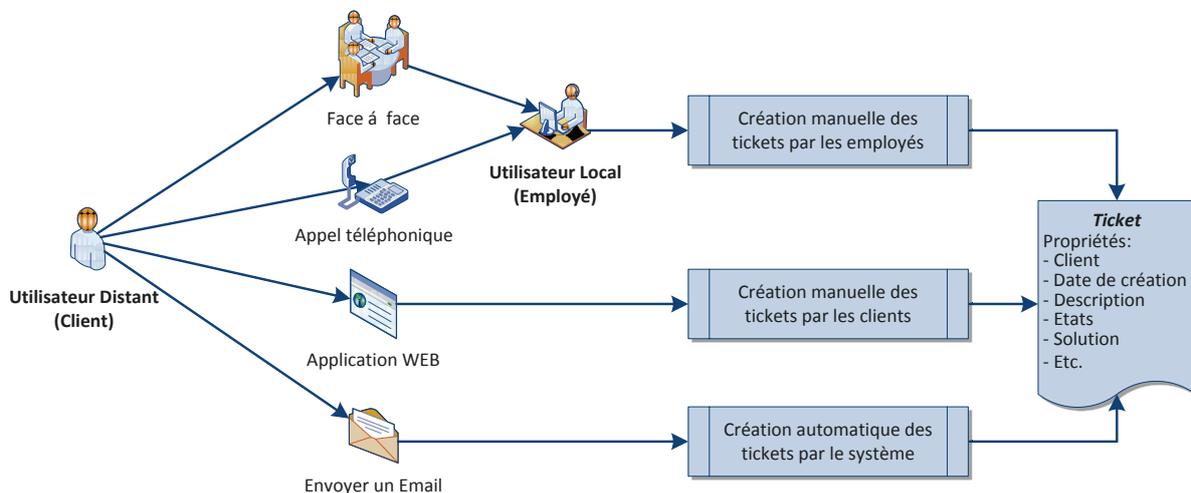


Figure 9 : Différentes méthodes pour créer un ticket dans B-WARN SUITE.

Le système couvre les 4 canaux de communications les plus utilisés par les clients : Face à face, Téléphone / Fax, Email et Web (voir Figure 9). En effet, la création des tickets dans le système est réalisée par les acteurs en utilisant trois méthodes différentes offertes par les deux applications du système :

- a- Méthode directe manuelle, à l'aide de B-WARN
 Cette méthode est exploitée par les utilisateurs locaux, en se servant de l'application bureautique, pour créer les tickets reportés par les clients d'une façon directe, via appel téléphonique (implicitement par fax) ou face à face.
- b- Méthode indirect manuelle, à l'aide de LIVE B-WARN
 Méthode utilisée par les utilisateurs distants pour créer leurs propres tickets en accédant, via internet, à l'application web du système.
- c- Méthode indirecte automatique, à l'aide de B-WARN
 Utilisée par le système, cette méthode sert à transformer en des tickets du système les emails envoyés à une ou plusieurs adresses électroniques spécifiques.

Un ticket une fois créé entre dans le processus de management du système, qui assure sa transition d'un état prédéfini à un autre. Il sera assigné à un employé pour le traiter comme il pourrait être escaladé d'un niveau de service à un autre, jusqu'à trouver la solution souhaitée. Durant le traitement, plusieurs emails pourront être envoyés et récupérés par le système et seront tous sauvegardés dans la base de données.

La fonction d'escalade des tickets de suivi assure les transitions des tickets dans les différents niveaux de service du département de service client afin d'aboutir à une meilleure répartition des tâches en fonction de leurs complexités et des compétences nécessaires pour les résoudre. Ce département qui a été restructuré au début de l'an 2009, avant le commencement du développement du projet, et organisé en une structure multi-tiers qui comporte trois niveaux de service : Helpdesk, Service technique et Service avancé (Chapitre I : L'entreprise). La figure suivante (Figure 10) résume le processus de d'assignation d'un ticket et son escalade d'un niveau de service à un autre.

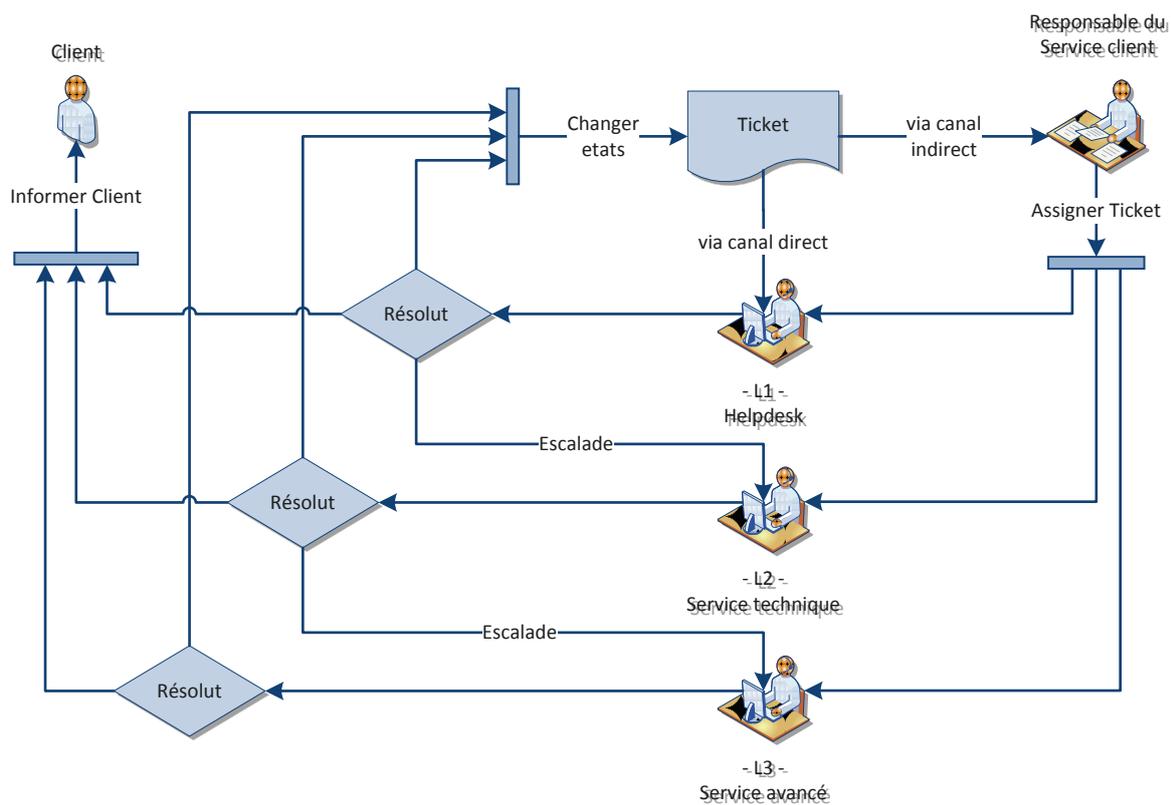


Figure 10 : Processus de traitement d'un ticket et son escalade entre les différents niveaux de services.

En plus, le système offre à l'administration de l'entreprise une large gamme d'outils statistiques qui aident à surveiller la performance du personnel du département de service client, et le temps de service approximatif offert à chacun des clients.

2. Choix des technologies

Les ressources matérielles existant déjà à PIXEL étaient toutes prêtes à être utilisées durant le développement de ce projet. Cette disponibilité de ressources matérielles représentait un avantage majeur pour l'équipe, mais un inconvénient pour la conduite de projet parce que l'utilisation de ses ressources était une obligation plutôt qu'un choix.

Donc, l'utilisation des technologies n'était pas un vrai choix, et cela pour plusieurs raisons :

- La minimisation du coût du projet
Etant un projet interne à l'entreprise et qui n'aurait pas un rendement financier direct, l'administration a décidé d'optimiser le coût du développement en commençant par l'utilisation des ressources existantes.
- Le langage de développement familier à l'équipe
Vu la courte durée accordée au développement de B-WARN SUITE, une modification du langage de programmation familier aux membres de l'équipe ou des technologies familières n'était pas une question ou une option suggérée.
- La librairie persistante à PIXEL
La grande librairie persistante dans l'entreprise fut une raison essentielle de développer le projet au lieu d'acquiescer un progiciel d'un fournisseur externe. Afin de profiter de cette librairie, l'utilisation des mêmes technologies de cette dernière était une obligation.

L'application bureautique du projet est développée en utilisant le langage de programmation VB.NET, tandis que l'application Web est développée en ASP.NET. De même, la base de données centralisée est créée en SQL Server. Durant le développement, plusieurs outils et bibliothèques ont été utilisés ; certains pour faciliter le travail des développeurs et la coordination, d'autres pour enrichir les interfaces utilisateurs etc. Dans ce qui suit, une description de ces ressources qui met en lumière leurs avantages et le but de leurs utilisations.

a- Microsoft SQL Server Management Studio 2008

SQL Server Management Studio est un environnement intégré qui permet d'accéder, de configurer et d'administrer tous les composants de SQL Server. Il a été utilisé pour créer et administrer la base de données relationnelle du projet. La version 2008 était commercialisée le 6 Août 2008. Elle soutient plusieurs formats de données structurées et semi-structurées, y compris des formats numériques pour les images, les audio, les multimédia et autres. En plus, SQL Server 2008 peut stocker une variété de données: XML, Email, fichier, documents etc.

b- Microsoft Visual Studio 2008

C'est le logiciel de développement utilisé pour développer les deux applications du projet (Bureautique et Web). Ce logiciel est un ensemble complet d'outils de développement permettant de générer des applications Web ASP.Net, des services Web XML, des applications bureautiques et des applications mobiles. Il utilise un environnement de développement intégré IDE (Integrated Development Environment), qui permet de partager des outils et de faciliter la création des solutions faisant appel à plusieurs langages.

La version 2008 est sortie en version finale à la fin de l'année 2007. Elle est fondée sur le .NET Framework 3.5. Elle comporte des outils de métrologie relatifs au code (indicateurs du nombre de lignes, profondeur des routines etc.) et à la performance (mémoire utilisée, temps d'exécution etc.). De plus, elle offre une meilleure gestion de test avec la possibilité d'élaborer des scénarios de test et des modules de test spécifiques aux applications AJAX.

c- Microsoft SourceSafe 6.0 (VSS)

Microsoft SourceSafe est le système de gestion des versions de développement du projet. Ce système sert à organiser le travail simultanément sur plusieurs versions des projets. Cette dernière fonction permet de conserver des versions de code en parallèle afin de pouvoir contrôler les modifications du code et de lutter contre les pertes accidentelles des fichiers. Il rend plus facile et plus intuitive la coordination au sein du projet.

d- Memory Profiler 3.0

Développé par SciTech Software, Memory Profiler est un outil qui sert à trouver les fuites de mémoire et d'optimiser l'utilisation de mémoire des logiciels écrits en C#, VB.NET et tous autres langages de programmation .NET.

Cet outil a été utilisé par l'équipe de développement pour contrôler et minimiser l'utilisation de la mémoire, durant l'exécution des fonctions complexe de B-WARN afin d'optimiser les surcharges de mémoire qui ralentissent et peuvent nuire au fonctionnement du système.

e- Dotfuscator Professional Edition 4.2

Dotfuscator Professional Edition est un outil qui sert à protéger les applications des risques de vol de propriété intellectuelle, d'effraction électronique et de piratage. Il est une solution d'obfuscation, de compactage et de filigranage qui sert à renforcer la protection des programmes basés sur le Framework .NET, contre la rétro-ingénierie (Reverse Engineering). Dotfuscator, développé par PreEmptive Solutions, est conçue pour empêcher les décompilateurs de produire une sortie utile et protéger les applications développées. Il est classifié comme outil d'obscurcissement des applications basées sur le Framework .NET.

Le filigranage est utilisé pour incorporer discrètement des données telles que des informations de copyright ou des numéros d'identification uniques dans les applications .NET. L'algorithme de filigranage utilisé par Dotfuscator n'augmente pas la taille des applications. En effet, il n'introduit pas de métadonnées supplémentaires qui pourraient nuire au fonctionnement.

f- Crystal Reports

Crystal Report est utilisé pour créer et afficher tous les rapports de B-WARN SUITE. Il est un composant de Visual Studio qui permet de créer des rapports aux contenus interactifs et soigneusement présenté dans l'environnement Windows. Les rapports complexes créés avec Crystal Reports peuvent être connectés à n'importe quelle source de base de données ou à des objets de données comme un DataSet ADO.NET. Les rapports hébergés dans des applications Web ou Windows sont affichés avec plusieurs visualiseurs spéciaux offerts par Crystal Report. La présentation des rapports est très interactive et offre des fonctionnalités telles que la navigation dans les pages, la recherche des textes et plusieurs autres fonctions.

g- NetAdvantage for .NET 2008 vol. 3 CLR 2.0

Offerte par Infragistics, la librairie NetAdvantage contient des composants interfaces utilisateurs pour Windows Forms, WPF (Windows Presentation Foundation), ASP.Net et Silverlight.

Cette librairie fréquemment utilisée par PIXEL dans le développement de ses différents logiciels, a été de même utilisée pour enrichir les interfaces utilisateurs des deux applications du projet B-WARN et LIVE B-WARN.

3. Cycle de vie et planification

Comme tout logiciel, ce projet possède des étapes de développement intégrées dans un cycle de vie. Le cycle de vie de ce projet a comporté toutes les étapes traditionnelles du développement, à l'exception de l'étape de documentation, ajournée du fait que ce projet sera utilisé par ses développeurs et leurs collègues. Les étapes du développement inclus dans le cycle de vie de B-WARN SUITE sont les suivants :

- | | |
|---------------------------------------|--------------------|
| 1. Définition des objectifs | 6. Tests unitaires |
| 2. Analyse des besoins et faisabilité | 7. Intégration |
| 3. Conception générale | 8. Qualification |
| 4. Conception détaillée | 9. Mise en service |
| 5. Codage (Programmation) | |

Le modèle choisi pour le cycle de vie de B-WARN SUITE a été le modèle en « V ». Ce modèle qui permet d'anticiper et d'identifier très tôt les éventuelles évolutions des besoins. Il permet de commencer plus tôt les plans de tests de qualification et d'évaluation des performances. De même, ce modèle permet de développer parallèlement les différents modules ou tâches après la validation de la conception générale.

La première étape du cycle de vie, la définition des objectifs, réalisée par l'administration de PIXEL, a fixé le développement d'un système de gestion de la relation client (GRC) comme objectif principale pour le projet. Toutefois, arrivée à la deuxième étape, l'administration a limité son objectif au développement d'un système de gestion du service client, du fait que ce dernier a été un besoin pressant pour l'entreprise et le temps de développement nécessaire pour un système GRC n'a pas été disponible.

Après la spécification des besoins et le dégagement des objectifs du produit souhaité, une planification chronologique détaillée a été réalisée. Cette planification a fixé le premier du Mars 2009, date du début des travaux, et fin d’Août 2009 comme date limite. Les heures de travail ont été calculées de la manière suivante :

- 1 jour de travail équivaut à 5 heures de travail effectif, puisque les membres de l’équipe étaient chargés d’autres tâches de service client et de développement.
- 1 semaine de travail équivaut à 6 jours de travail (samedi inclus).

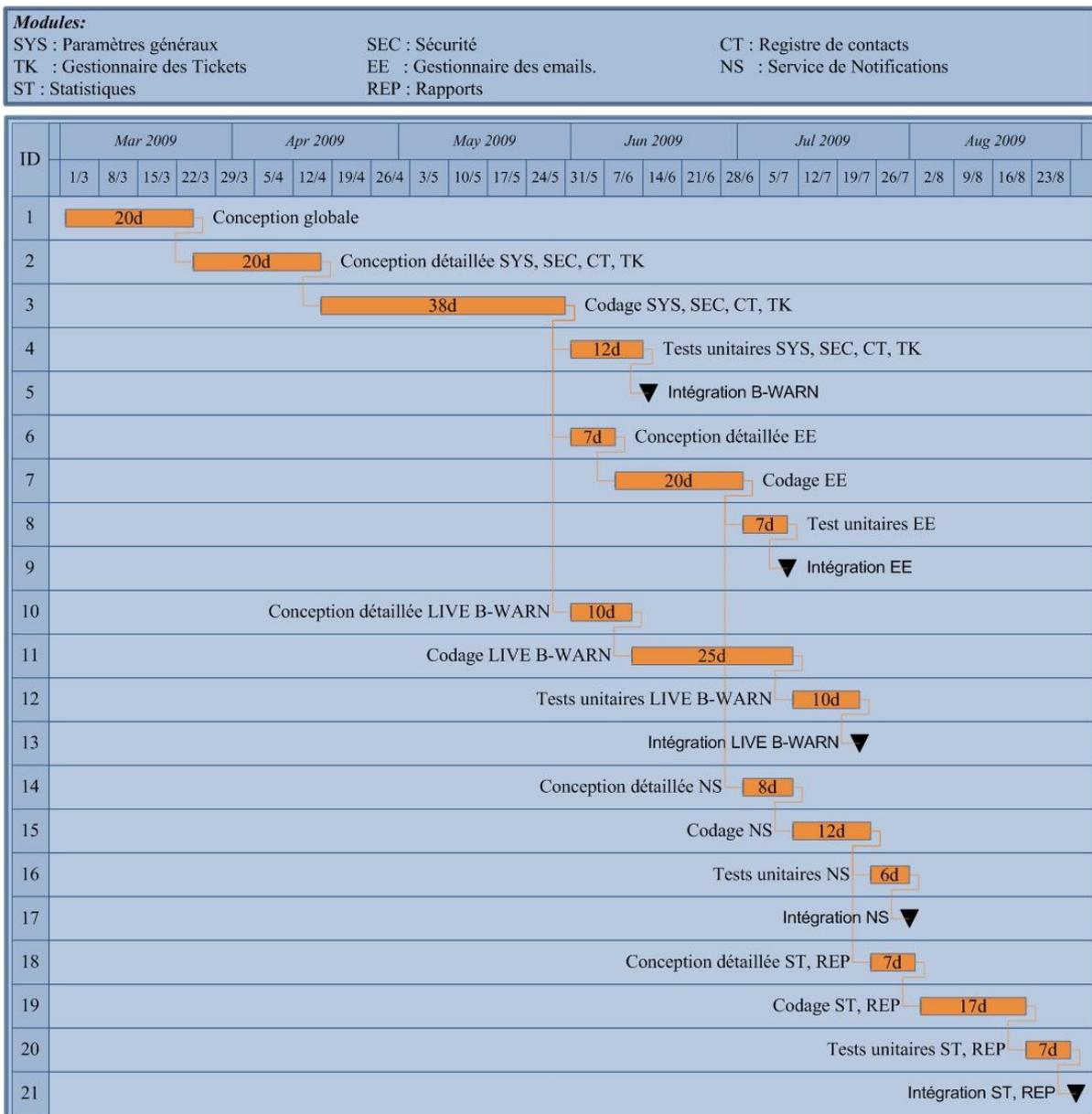


Figure 11 : Diagramme de GANTT du développement du projet.

Le diagramme de GANTT affiché par la Figure 11 décrit les différentes tâches de développement du projet, leurs dates de début, leurs dates de fin, et la durée de chacune. De plus, ce diagramme visualise les dépendances entre ses différentes tâches de développement.

La première étape de cette planification, la conception générale, a été réalisée par tous les membres de l'équipe de développement. Elle a comporté plusieurs recherches sur les solutions disponibles au marché, similaires au système en cours de développement. Ces recherches ont abouti à une formulation des fonctionnalités de B-WARN SUITE, et à une vision pour sa prochaine version (voir Chapitre VI : Solutions disponibles).

La planification a notamment distingué entre le développement des deux applications du projet. B-WARN étant l'application principale du projet, le développement de ses composants essentiels était prioritaire, par rapport à celui des composants de LIVE B-WARN.

De même, elle a distingué entre les priorités des différents modules de B-WARN, l'application bureautique, et leurs dépendances afin de commencer le plus tôt possible les phases des tests de fonctionnalités des modules initiales en parallèle avec le développement des autres modules secondaires.

La planification a comporté cinq dates d'intégration consécutives qui sont les dates de la fin du codage et des tests unitaires primaires de chaque module ou application, et le début de son utilisation par les autres membres du personnel qui ne participent pas au développement. Celui-ci considéré comme étant une deuxième phase de tests des fonctions intégrées.

Les différentes tâches de développements sont réparties sur les membres de l'équipe de développement. Cette répartition a pris en considération les compétences et les qualifications de chaque membre et a visé à développer en parallèle le plus grand nombre possible de ses tâches afin de profiter au maximum du temps.

4. Equipe et répartition des tâches

Au début du projet, l'administration de l'entreprise lui a accordé une grande priorité, vu son importance et son influence sur l'avenir de l'entreprise.

Une équipe de quatre personnes a été composée pour développer cette première version du projet. Moi, étant membre de cette équipe et ayant alors décidé de devenir le responsable du département du service client, j'ai été nommé chef de ce projet.

L'expérience professionnelle des membres de l'équipe était disproportionnée. Deux membres avaient entre trois et quatre ans d'expérience professionnelle, et les deux un an seulement. Pour cela les tâches du projet étaient réparties selon leurs types ; les tâches d'analyse et de recherches ont été assignées aux membres les plus âgés professionnellement et aux autres membres celles de programmation qui ne nécessite qu'une connaissance technique de développement. Mais durant les phases de conception, parfois la participation de tous les membres de l'équipe a été recommandée et parfois celle des autres membres de PIXEL.

Les membres de l'équipe de développement du projet ont été chargés d'autres tâches de développement dans d'autres projets au cours de développements à PIXEL, et des tâches de service client. Cette situation a empêché d'avoir une équipe qui travaille à plein temps dans ce projet. Ceci a demandé l'aide de tout le personnel de l'entreprise, selon ses disponibilités, surtout durant les phases de test des fonctions.

Un mois après le début du développement, à cause de l'émigration inattendue du membre le plus âgé et le plus qualifié de l'équipe de PIXEL, j'ai été libéré de ma nouvelle position pour remplir le vide et prendre la responsabilité d'une implémentation d'IRIS dans une grande société internationale située à Dubaï. Cette nouvelle situation a affecté d'une façon directe la progression du projet (voir Chapitre IX : Conduite du projet).

Chapitre VIII : B-WARN SUITE

Ce chapitre comporte une description détaillée de B-WARN SUITE, ses applications, ses composants, ses modules et ses fonctions essentielles. De même, il englobe plusieurs diagrammes UML (Unified Modeling Language) qui servent à mieux comprendre les fonctionnalités et le flux de travail (Workflow) du système.

1. Paramètres généraux

Ce projet est développé de telle sorte qu'il pourra être personnalisé pour différentes entreprises qui comportent un service client, selon leurs propres types d'affaire et leurs règles et méthodes de management. Cette fonctionnalité a exigé la création de plusieurs paramètres modifiables qui seront spécifiés par l'administrateur de l'entreprise, afin de définir le comportement du système.

Dans ce qui suit une définition des principaux paramètres de base qui devront être spécifiés avant de faire démarrer le système :

- *Informations générales*

Elles représentent les informations générales de l'entreprise : nom et coordonnées (adresses numéros de téléphone etc.). Ces données seront affichées par différents interfaces et rapports du système.

- *Les niveaux de service*

Ces niveaux définissent la structure du département du service clientèle de l'entreprise : Helpdesk, service avancé et Ingénieur de service ou Niveau 1 Niveau 2 etc. Cette définition permet de regrouper les employés ou les utilisateurs locaux de B-WARN afin de pouvoir escalader les tickets d'un niveau à un autre en arrangeant ces niveaux selon un coefficient, associé à chacun, qui décrit l'hierarchie des niveaux.

- *Les canaux de communication*

Ils représentent les différents canaux de communication avec les clients : Téléphone, Email etc. Ces canaux seront utilisés pour indiquer d'une part la méthode par laquelle les tickets sont créés dans le système, d'autre part le canal utilisé pour envoyer la solution au client ou la réponse à sa requête.

- *Les projets*

Les projets sont les produits de l'entreprise et spécifiquement ceux qui nécessitent un service après vente et par le fait même une interaction avec le service client. Ces projets sont utilisés en créant les tickets, afin de lier les requêtes aux produits convenables.

- *Les catégories*

Ces catégories servent à regrouper les tickets du système selon le besoin de l'entreprise et son type d'affaire. Par exemple, pour les logiciels, les catégories peuvent être : Erreur de programmation, Erreur de base de données, Question etc.

- *Les sévérités*

Ces sévérités indiquent le degré de dégât procuré par l'erreur : Bloqueur, Critique, Mineur, Trivial etc. Ce paramètre sert à définir la vitesse de traitement du ticket par rapport aux autres tickets.

- *Les priorités*

Les priorités sont les différents niveaux de classification des clients, des projets et des tickets du système. Cette classification peut comprendre une ou plusieurs classes selon le besoin de l'entreprise. A chaque priorité sera agrégé un poids qui sert à identifier son classement par rapport aux autres, et plusieurs autres paramètres graphiques qui servent à afficher les tickets d'une manière vive et mieux compréhensible (Figure 12).

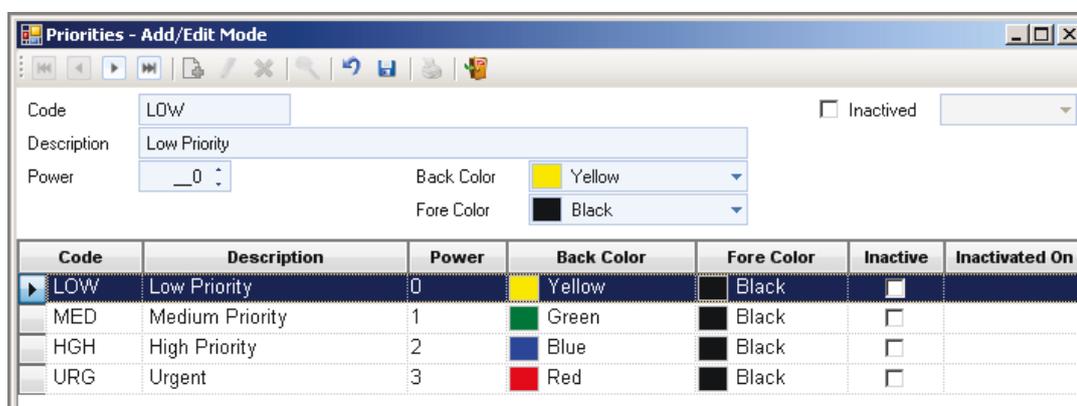


Figure 12 : Fenêtre de saisie des priorités dans B-WARN.

2. Interprétation des composants

Comme c'est déjà mentionné, ce système est composé de deux applications physiquement séparées une B-WARN, l'application bureautique, et LIVE B-WARN, l'application Web. Dans ce qui suit une description détaillée des deux applications, leurs composants et leurs fonctions principales.

2.1 B-WARN

L'application bureautique nommée B-WARN est l'application essentielle du système. Elle est conçue pour gérer toutes les données concernant les clients, les tickets, la gestion des accès. De même, cette application incorpore toutes les fonctionnalités du système. Elle est complètement indépendante de la deuxième application LIVE B-WARN.

Lors du lancement de B-WARN, et après authentification de l'employé, l'interface principale de l'application s'ouvre et affiche toutes les fonctions autorisées à l'utilisateur. Le menu principal sera affiché en même style que le menu d'OUTLOOK. Ce style a été choisi parce que sa façon d'affichage en accordéon permet d'insérer un grand nombre de liens dans une espace minime, sans oublier son comportement dynamique et vif.

Les liens comportés par ce menu sont regroupés en différentes sections qui symbolisent les différents modules du système.

De même, cette interface affiche plusieurs autres composants comme la barre d'outils (Tool bar) qui contient des raccourcis (Shortcuts) des fonctions et des fenêtres des plus utilisées, et la barre d'état (Status bar), qui visualise l'état de connectivité du système, le nom de l'utilisateur en connexion et d'autres informations utiles.

L'interface principale est divisée graphiquement en six zones. La Figure 14 montre les différentes zones de cette interface et décrit le contenu affichés de chacune.

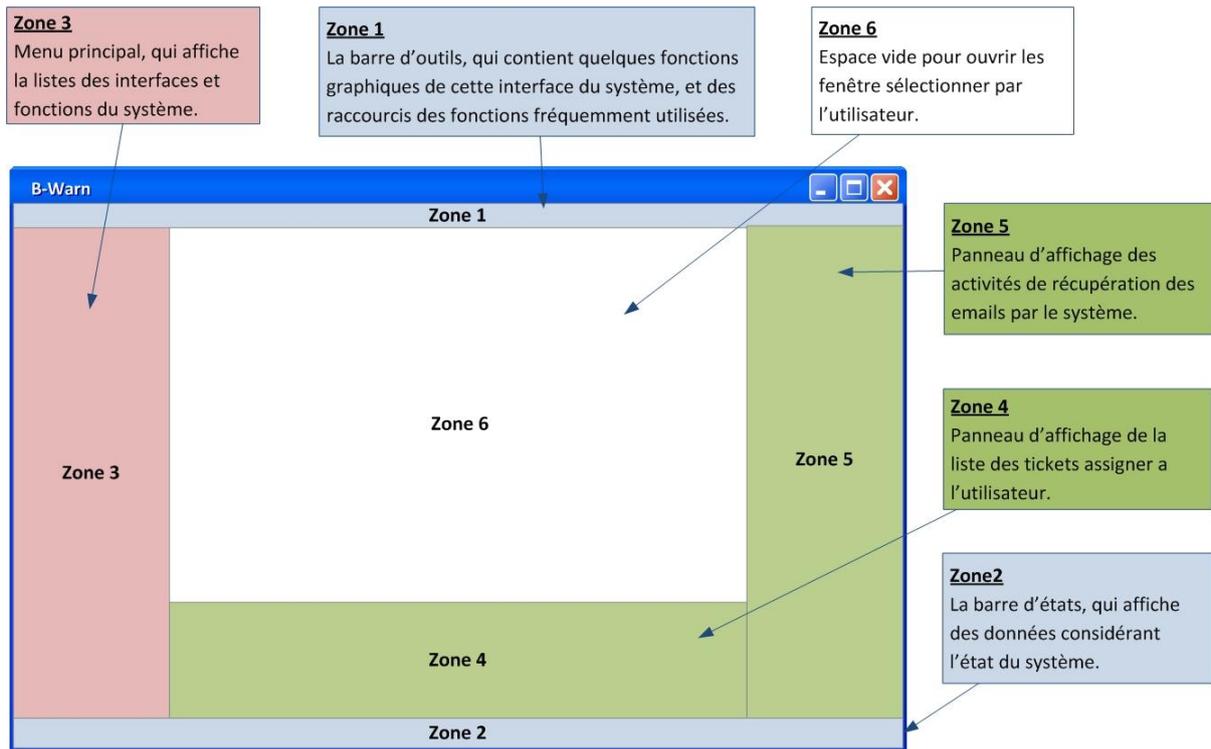


Figure 14 : Composition graphique de l'interface principale de B-WARN.

Les zones 1 et 2 sont des parties fixes de l'interface tandis que les zones 3,4 et 5 sont mobiles et peuvent être modifiées par l'utilisateur lui-même.

Les zones 4 et 5 sont reliées à des paramètres spéciaux du système qui précisent si les données affichées par ces zones sont récupérées soit automatiquement, dans un intervalle de temps fixé par l'administrateur du système, soit manuellement en appuyant sur les boutons de rafraîchissement situés dans chacune des ces zones.

2.1.1 Les différents module de B-WARN

Le menu affiché par la zone 3 de l'interface principale comprend les liens de toutes les fenêtres et les fonctions de l'application. Ce menu comprends huit parties qui correspondent chacune à un des modules de cette application. Dans ce qui suit une description détaillée de chaque module et des principales fonctions intégrées par chacun.

A. Paramètres du système

Ce module est responsable de la gestion de tous les paramètres généraux du système expliqués au début de ce chapitre. Ces paramètres sont la base du système. Ils sont

personnalisés selon les standards et la méthode de management de l'entreprise qui utilise B-WARN, en utilisant les différentes fenêtres de saisie offertes par ce module. Parmi tous les autres, ce module est donc le premier à être utilisé.

B. Sécurité

Ce module gère les accès et les règles de sécurité du système. Il est responsable d'identifier et de valider les utilisateurs du système et de contrôler leurs accès aux différentes interfaces, rapports et fonctions du système, selon les permissions attribuées par l'administrateur du système pour chacun d'eux. Deux types d'accès sont gérés par ce module : Les accès locaux des employés à B-WARN et les accès distant des clients à LIVE B-WARN.

Les Accès Locaux

Les utilisateurs du B-WARN sont enregistrés d'une manière simple en utilisant une seule fenêtre de saisie. Pour chaque utilisateur devront être spécifiées plusieurs données nécessaires, et toutes les permissions d'accès aux différentes fonctions du projet.

Les paramètres nécessaires à créer un utilisateur local sont les suivants :

- Un code d'utilisation et un mot de passe forment ensemble l'identité de chaque utilisateur.
- Le niveau de service de l'entreprise auquel il appartient. Ce niveau serait utilisé pour regrouper les utilisateurs en vue d'appliquer la fonction d'escalade des tickets d'un niveau de service à un autre.
- L'adresse électronique avec tous ces paramètres de connexion (nom des serveurs SMTP et POP3 etc.) utilisée pour envoyer des emails à l'aide du B-WARN.
- La signature des courriers électroniques envoyés par l'utilisateur.

Deux types d'autorisations sont gérés pour les utilisateurs locaux : les accès aux interfaces et rapports de l'application et les accès aux fonctions de manipulation des tickets de chaque projet de l'entreprise. La combinaison des deux types fournit à l'entreprise un grand éventail de limitations de ces données.

Les accès aux interfaces de l'application sont définis par quatre types de permission :

- Permission de voir, qui autorise l'utilisateur de voir les données affichées par l'interface.
- Permission de créer / modifier / supprimer. Ce sont les permissions traditionnelles qui autorisent l'utilisateur à manipuler les données affichées par l'interface.

Tandis que les accès aux tickets des projets sont définis par six types de permissions :

- Permission de voir, qui autorise l'utilisateur de voir (implicitement imprimer) les tickets du projet.
- Permission de créer/modifier/supprimer. Elle autorise l'utilisateur à créer modifier ou supprimer les tickets du projet.
- Permission d'assigner, qui autorise l'utilisateur à assigner ou de réassigner les tickets du projet aux différents utilisateurs du système.
- Permission d'escalader, autorise l'utilisateur à escalader les tickets d'un niveau de service à un autre.

Pour les rapports statistiques, un seul type de permission est valable : permission d'exécution.

Les Accès distants

Les utilisateurs distants du système sont créés d'une façon simple. Ces utilisateurs devront être inscrits dans le registre des contacts du système et certainement liés à une entreprise cliente. Un mot d'utilisation et un mot de passe seront associés à ces contacts afin de pouvoir les identifier en accédant LIVE B-WARN. En plus, des permissions d'accès seront associés à chaque utilisateur afin de limiter son accès aux différentes pages et fonctions de l'application Web.

Les permissions de ces utilisateurs sont spécifiques pour chacune des fonctions de l'application Web qui seront expliquées au cours de ce chapitre.

Une limitation d'affichages des tickets et aussi valable pour ces utilisateurs. En cas de limitation, l'utilisateur ne pourra voir et manipuler que les tickets qu'il a rapportés, sinon il pourra manipuler tous les tickets de son entreprise.

Le suivant diagramme des classes schématise les différentes classes et entités de ce module.

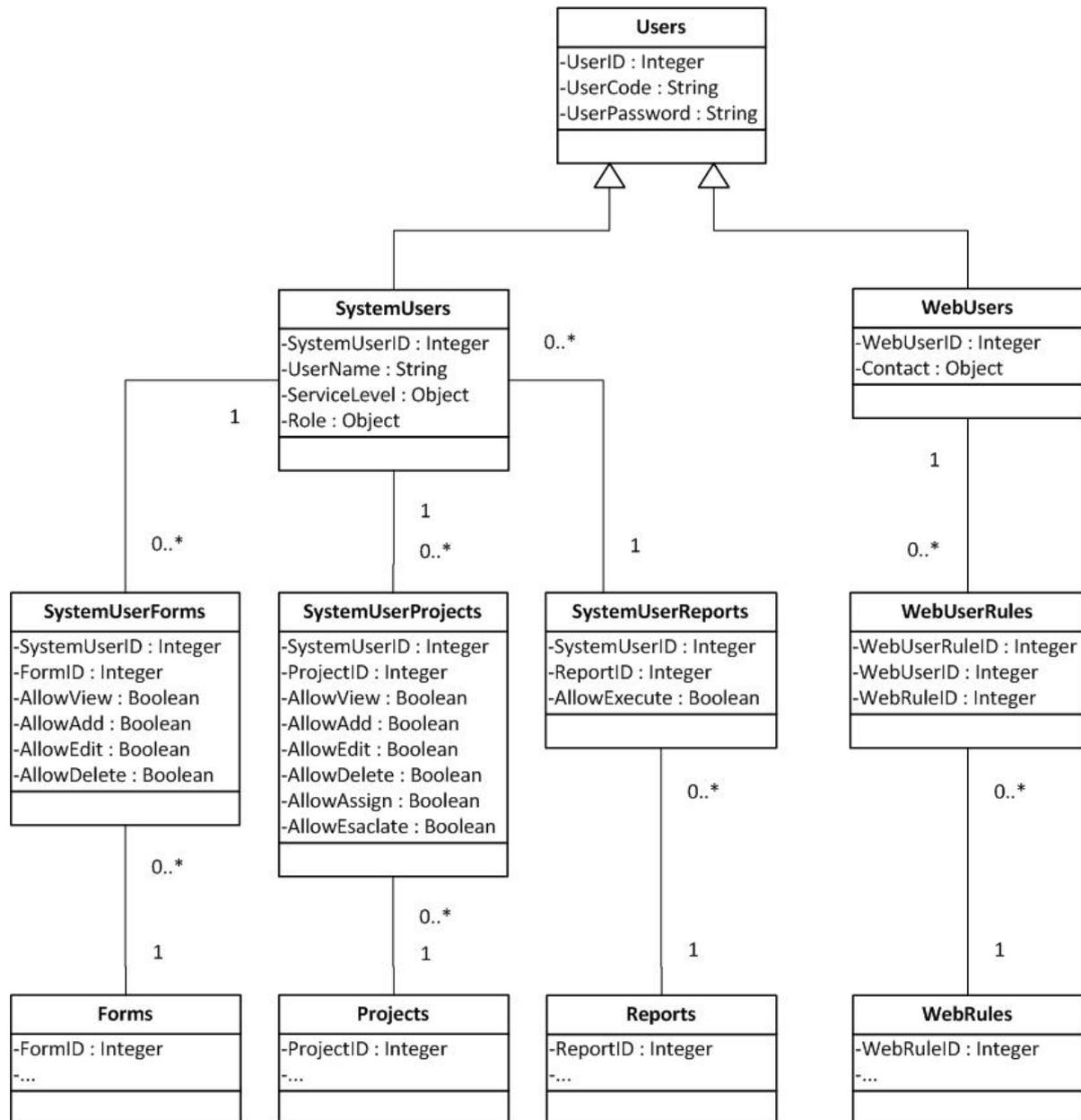


Figure 15 : Diagramme de classes du module de sécurité.

C. Registre des contacts

Le registre des contacts est le module responsable de la gestion de toutes les données des clients et de tous leurs employés en contact avec le service client de l'entreprise qui utilise ce système.

Dans ce module, chaque entreprise cliente est représentée par une entité nommée *Compagnie* et chacun de ces employés est représenté par une autre entité nommée *Contact*.

Plusieurs données sont sauvegardées dans le système, pour chaque compagnie créée :

- Les coordonnées
Toutes les coordonnées utiles seront spécifiées pour chaque client : les adresses les numéros de téléphone et les adresses électronique.
- Les SLA (Service Level Agreements) ou contrats de niveau de service
Ces contrats définissent le niveau de service qui devrait être offert au client. Ces SLA contient les données nécessaires à la gestion des tickets du client :
 - Les dates du début et de la fin du contrat, qui spécifient la période de service.
 - Le projet acquis par le client, et qui doit être couvert par le service client
 - Le délai maximal de réponse pour chaque catégorie de ticket.
 - La priorité ou l'importance du client par rapport aux autres clients.
 - L'employé de service auquel seront assignés les tickets du client (optionnel).

Pour chacune des compagnies enregistrées, plusieurs contacts pourront être créés. L'entité qui correspond à chaque contact contient les données suivantes :

- Le prénom et le nom
- La photo, si elle existe
- Le titre social et le titre professionnel
- La date et le lieu de naissance
- Les coordonnées (Adresses, numéros de téléphone et adresses électronique)

Les coordonnées créées pour les compagnies et les contacts sont reliées à des types définis par l'administrateur du système, ce qui permet de créer plusieurs coordonnées selon le besoin de l'entreprise. Par exemple :

- Les types d'adresses : Adresse du domicile, Adresse au travail etc.
- Les types de téléphone : Téléphone au bureau, Téléphone mobile etc.
- Les types d'adresses électroniques : Courriel personnel, Email Professional etc.

Les adresses électroniques sauvegardées sont les plus importantes au système des coordonnées. Elles sont considérées comme clés d'identification des contacts et des compagnies. Elles seront utilisées en récupérant les courriers électroniques afin d'identifier

l'envoyeur de ces courriers, donc identifier le client, sa priorité et toute autre information nécessaire.

Les différentes fonctions de ce module sont résumées dans le diagramme de cas d'utilisation, illustré dans la Figure 16.

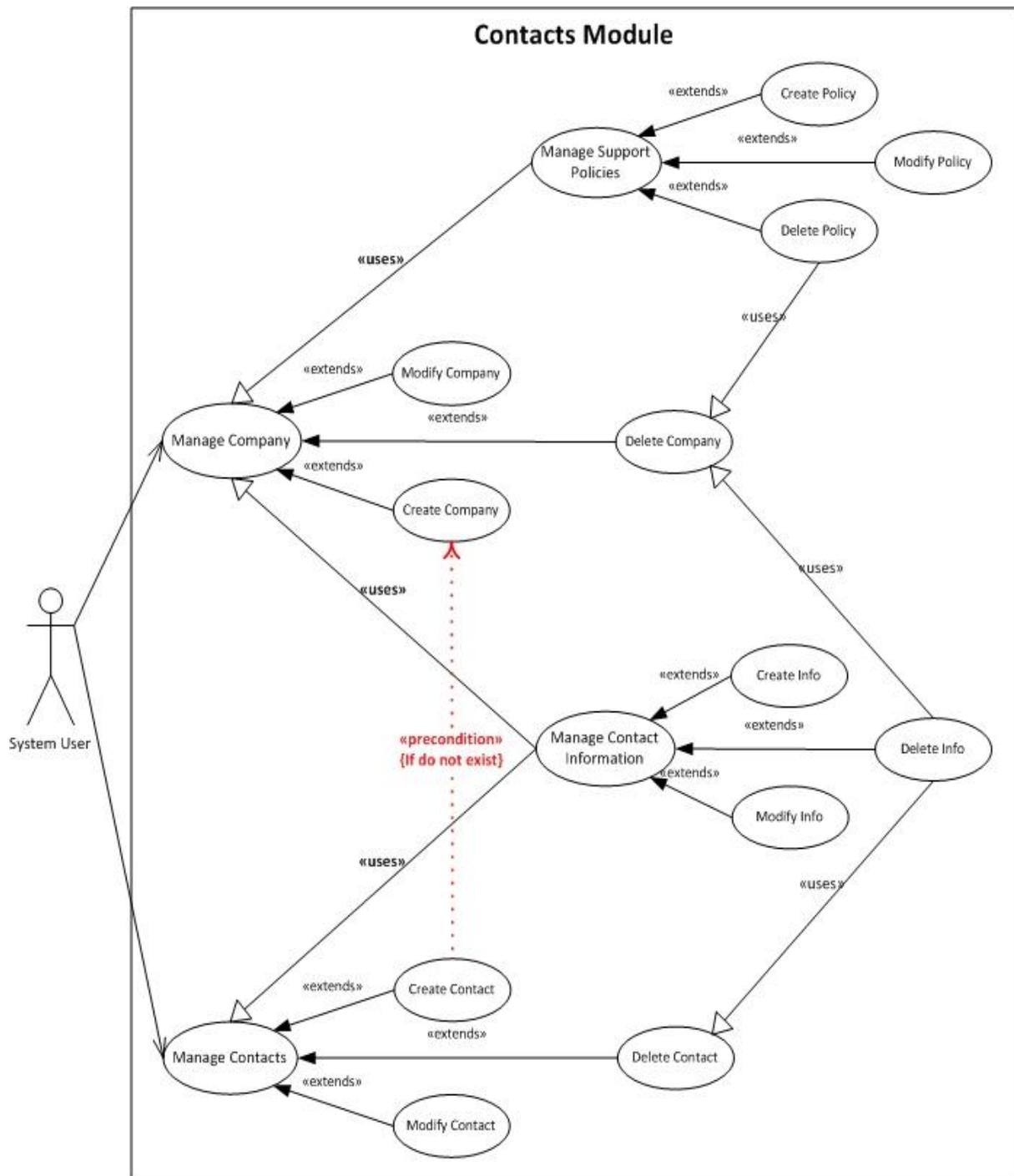


Figure 16 : Diagramme de cas d'utilisation du module "Registre des contacts".

D. Gestionnaire des Tickets

Considérer comme module essentiel du système, ce module est conçu pour traiter les tickets du système, et assurer leurs transitions d'un état à un autre, selon le cycle de vie des tickets incorporé dans B-WARN. Il offre plusieurs fonctions de manipulation des tickets : Créer, Modifier, Assigner, Escalader etc.

Un ticket, qui représente un problème, une erreur ou la question d'un client, est créé par une méthode directe ou indirecte, d'une manière automatique ou manuelle, et initialisé à l'état *Neuf*. Devront être spécifiés pour chaque ticket, le client qui l'a rapporté, le projet auquel il est relié, sa catégorie et sa sévérité. D'après ces quatre variables, la priorité du ticket est sélectionnée en se référant au tableau de gestion des priorités défini dans les paramètres du système. Le ticket contiendra non seulement une description du problème mais aussi de plusieurs fichiers électroniques (fichier Excel, fichier PDF, image etc.) qui pourront être utiles à son traitement.

Après la création de chaque ticket, un numéro de série lui sera associé. Ce numéro représente la clé d'identité du ticket et sera utilisé comme référence dans toutes les correspondances avec le client. Le ticket sera assigné automatiquement à l'utilisateur spécifié dans le SLA du client. Il pourrait être modifié par l'administrateur ou n'importe quel autre utilisateur ayant le droit d'assigner les tickets du projet en question.

Du début et jusqu'à la fin du traitement, l'état de ticket sera fixé à *Ouvert*. Pendant le traitement, des informations supplémentaires pourront être nécessaires. L'état du ticket sera fixé à *Suspendu* en attendant ces dernières, et sera restitué à *Ouvert*, lors de la reprise du traitement.

Durant cette même période plusieurs, emails pourront être envoyés et reçus du client en utilisant le système, sans avoir besoins d'un autre logiciel. Tous ces emails seront sauvegardés dans la base de données et attachés au ticket.

A la fin du traitement du ticket, si la solution souhaitée est atténuée, l'état du ticket sera fixé à *Fermé*, sinon l'utilisateur (employé) demande d'escalader le ticket à un niveau de service supérieur, en spécifiant la cause de sa demande. Dans le deuxième cas, il sera crée une

demande d'escalade comportant une variable définissant son état. Cette dernière sera fixée à *Attente d'approbation*.

La demande d'escalade pourrait être *Refusée* ou *Annulée* par le responsable, ou elle pourrait être *Acceptée*. Le ticket sera aussi réassigné à un des employés des niveaux de services supérieurs et son état sera fixée à *Escalader*. Ces demandes sont intégrées dans le cycle de vie des tickets afin d'en éviter les escalades arbitraires et non justifiées.

Au cas où un ticket fermé nécessiterait une réconciliation de la solution proposée ou la recherche d'une autre solution, l'état du ticket sera fixé à *Rouvert* et le ticket sera traité comme un ticket à un état *Ouvert*.

La Figure 17 illustre le diagramme d'états qui résume le cycle de vie d'un ticket, ses transitions d'un état à un autre, et les actions provoquant ces transitions.

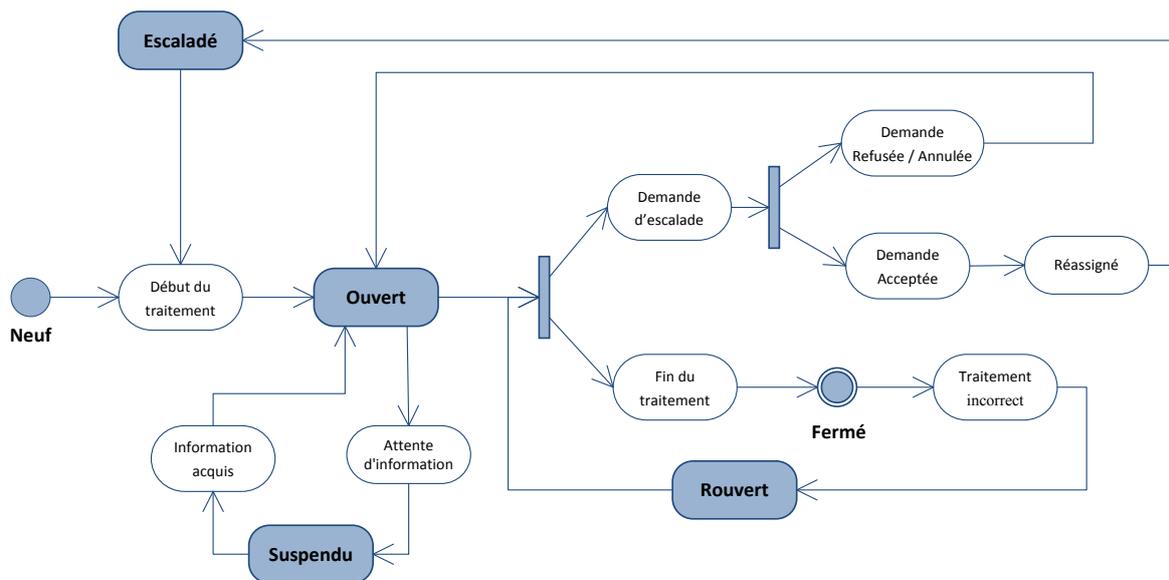


Figure 17 : Cycle de vie d'un ticket dans B-WARN.

Lors de la fin du traitement, les solutions proposées sont regroupées en plusieurs types définis par l'entreprise. Par exemple : Explication, Mise à jour du produit, Requête de données, etc. Ces types aideront plus tard à regrouper les modifications des différents projets de l'entreprise et à identifier les sources de problèmes des différents produits.

Chaque solution proposée et demande d'escalade possède un compteur qui serait remplie par l'employé pour spécifier le temps de traitement estimé. Cette variable sera utilisée pour évaluer le temps de service dépassé pour chaque ticket, client, projet etc.

Enfin, le système sauvegarde d'une part une historique détaillée de tous les états par lesquels un ticket est passé, et d'autre part la liste de tous les utilisateurs auxquels ce ticket a été assigné. De même, toutes les correspondances avec le client et toutes solutions proposées seront sauvegardées et attachées au ticket.

Le diagramme des classes affiché par la figure suivante schématise la structure des tickets et toutes les données qui lui sont attachées:

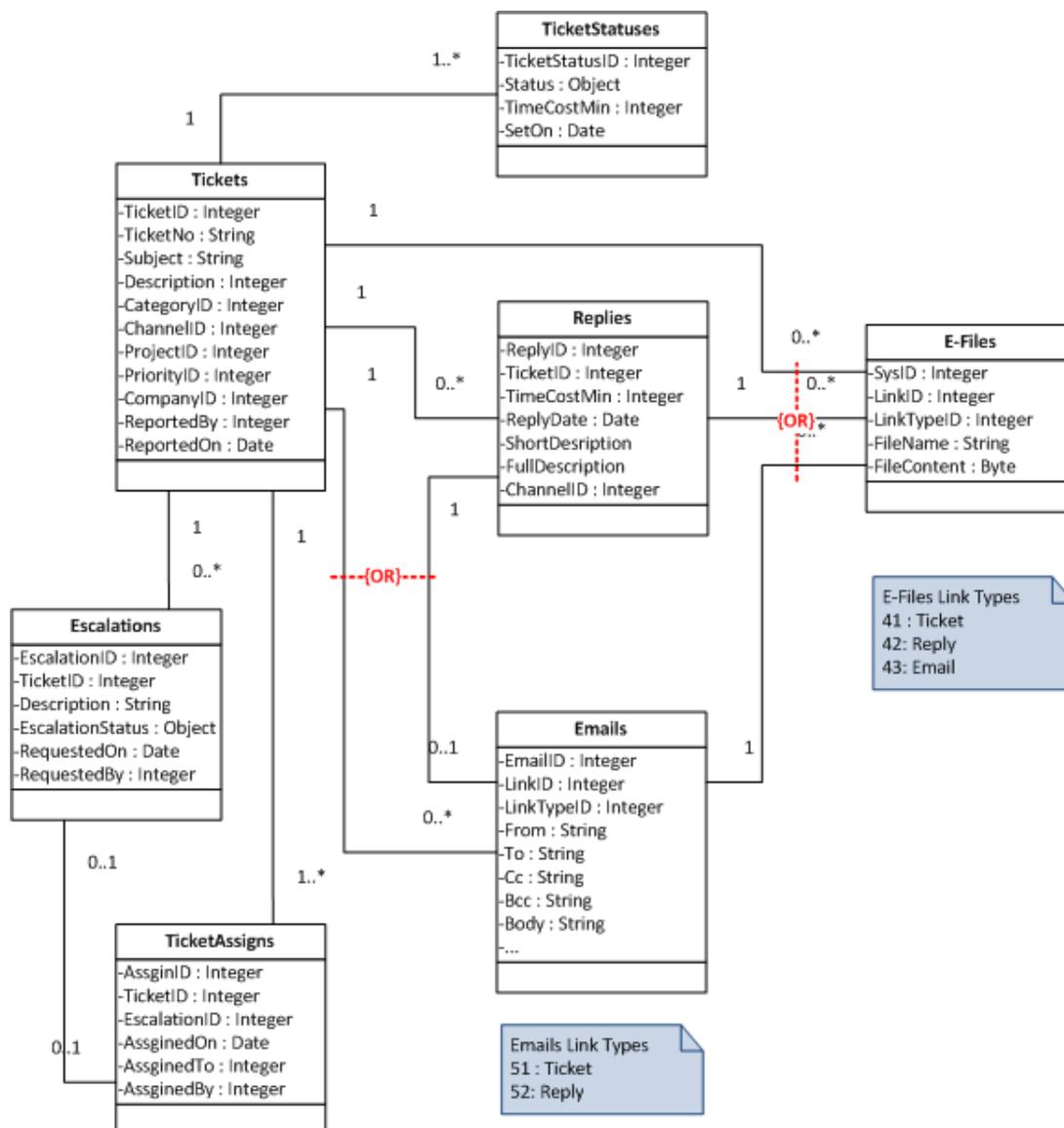


Figure 18 : Diagramme de classes du module "Gestionnaire des Tickets".

E. Gestionnaire des Emails

Ce module est l'agent responsable de l'intégration et de la manipulation des emails dans B-WARN. Il offre au système les fonctionnalités d'envoyer et de recevoir des courriers électroniques. Son rôle principal est de récupérer les emails envoyés par les clients, de les transformer en tickets de suivi ou de les lier aux tickets persistants déjà créés.

Plusieurs comptes de messagerie électronique seront enregistrés par ce module. Ces comptes seront employés par les utilisateurs du système pour envoyer des messages, comme ils seront utilisés pour récupérer les messages envoyés par les clients. Les comptes enregistrés dans le système possèdent plusieurs paramètres :

- L'adresse électronique et son nom d'affichage qui correspondent au compte.
- Le serveur SMTP et sa porte de connexion, qui seront utilisés pour envoyer les messages à travers ce compte.
- Le serveur POP3 et sa porte de connexion, qui seront utilisés pour la récupération des courriers électronique de la boîte de réception du compte. Une variable aussi devrait être définie qui spécifie si les messages seront supprimés du serveur POP3 après leur récupération ou non.
- Le mot d'utilisateur et mot de passe utilisés pour s'authentifier en accédant aux serveurs.
- Auto-synchronisation et son intervalle de temps, qui définissent si la boîte de réception du compte doit être réconciliée automatiquement, à chaque intervalle de temps, pour récupérer les nouveaux les messages.
- Réponse spontanée et contenue du message, qui incite le système à envoyer automatiquement un message aux envoyeurs des messages récupérés. Le message sera personnalisé pour chaque compte.

Ce module englobe toutes les interfaces traditionnelles de gestion des emails :

- La boîte de réception (Inbox), qui visualise tous les emails récupérés par le système.
- Les messages envoyés (Outbox), qui affichent tous les emails envoyés du système en utilisant les différents comptes.
- Les brouillons (Drafts), pour les messages écrits, non envoyés.

- Messages supprimés (Deleted messages), qui liste les messages supprimés du système. Ces messages pourront être détruits ou restitués.

Une interface particulière est désignée pour permettre aux utilisateurs de créer de nouveaux messages ou manipuler ceux qui existent, en utilisant les fonctions de manipulation classiques citées ci-dessous :

- Marquer comme lu / no lu (Mark as read / unread)
- Répondre / Répondre a tous (Reply / Reply to all)
- Transférer (Forward)
- Enregistrer un brouillon (Save as draft)

En outre, cette interface offre aux utilisateurs une fonction de recherche des adresses électroniques en consultant les coordonnées existant dans le registre des contacts.

Chaque message envoyé par B-WARN possède, comme adresse d'envoyeur, le compte email spécifié pour l'utilisateur lors de son inscription dans le système. De même, les messages comporteront la signature propre de l'utilisateur déjà spécifiée.

Les messages envoyés du système pourront être solitaires ou liés à un des tickets du système. Dans le deuxième cas, le numéro du ticket sera affiché dans le texte d'objet (ou titre) du message par le format suivant : [[NUMERO TICKET]]. Ce format sera utilisé afin de lier tous les emails envoyés des clients au ticket convenable, afin d'éviter la création de nouveaux tickets à chaque réception d'un nouveau message.

La récupération des messages pourrait être réalisée par deux méthodes:

A. Manuellement, à l'aide d'une interface d'assistance (Wizard)

Une interface d'assistance est conçue pour aider les utilisateurs autorisés à récupérer les nouveaux messages manuellement. Il suffit que l'utilisateur sélectionne les comptes à consulter de la liste des comptes visualisée par cette interface. Le système consultera par la suite ces comptes, l'un après l'autre afin de récupérer tous les nouveaux messages de leurs boîtes de réception.

B. Automatiquement, par le système

A l'aide d'un agent *Windows Services* installé et démarré sur une des stations du réseau de l'entreprise, le système pourra effectuer la synchronisation des comptes sans interventions des utilisateurs. Ce service consulte la base de données du système et récupère la liste des comptes qui ont la variable *Auto-synchroniser = VRAI*. Ensuite la récupération des messages de chacun de ces comptes sera réalisée.

L'historique des activités de récupération des emails, manuel ou automatisée, et leurs résultats sont sauvegardés par le système dans une table de données. Cette historique est affichée dans la Zone 3 de l'interface principale du B-WARN et également par une interface spéciale de ce module qui permet de manipuler ces données.

L'événement de consultation des boîtes de réception des comptes et de récupération des nouveaux messages est effectué par des fonctions qui assurent la connexion au serveur POP3 en utilisant les paramètres du compte de messagerie électronique en question, et qui vérifie si les nouveaux courriers de la boîte de réception sont sauvegardés dans la base de données du système.

La vérification de l'existence des emails se réalise en utilisant l'UID (Unique Identifier) qui est une clé internationale unique pour chaque message. Cette clé est sauvegardée dans la base de données pour éviter la duplication des courriers récupérés.

Après la récupération de chaque nouveau message, le système recherche dans son titre le format `[[NUMERO TICKET]]`, défini déjà, pour trouver un numéro de ticket qui persiste dans le système. Si un numéro de ticket trouvé et validé, le message sera lié à ce dernier pour éviter de créer plusieurs tickets pour la même requête.

En cas de récupération d'un email isolé, qui n'est pas relié à un ticket du système, et au cas où son expéditeur serait connue, un nouveau ticket sera créé en gardant tous les fichiers et images attachés et cela après détection de l'entreprise à laquelle appartient l'expéditeur et toutes les données nécessaires : projet, priorité, responsable de service etc.

Après la création d'un nouveau ticket, si l'option *Auto-Reply* du compte est sélectionnée, une connexion sera établie au serveur SMTP du compte et un message sera envoyé à l'expéditeur du courrier récupéré comportant le numéro du ticket créé.

Si son envoyeur est anonyme, le message sera enregistré dans le système sans ses fichiers électroniques joints, et il est marqué *Non lu* jusqu'à ce que l'un des utilisateurs autorisés décide de le supprimer ou de surcharger ses fichiers joints et le transformer manuellement en un nouveau ticket. Ceci permet d'éviter la réception des courriers indésirables ou les spam.

Le suivant diagramme de séquences (Figure 19) récapitule l'événement de récupération des messages d'un serveur POP3 et d'envoi des messages automatique à travers le serveur SMTP:

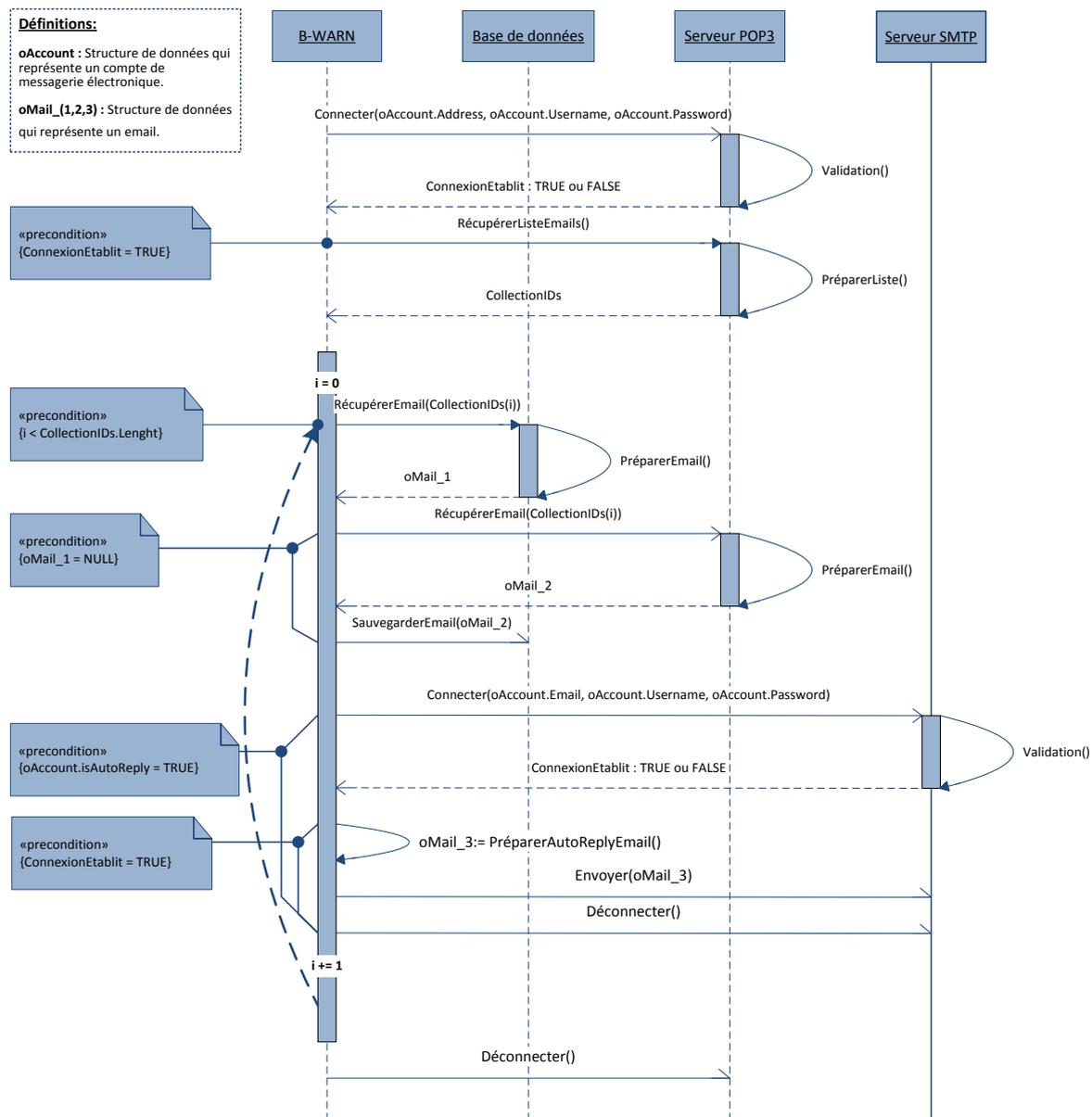


Figure 19 : Diagramme de séquences : La récupération des messages du serveur POP3 dans B-WARN.

Le processus de transformation des messages isolés en des tickets *Neufs* et de liaison des autres messages aux anciens tickets est schématisé par le diagramme d'activités de la Figure 20.

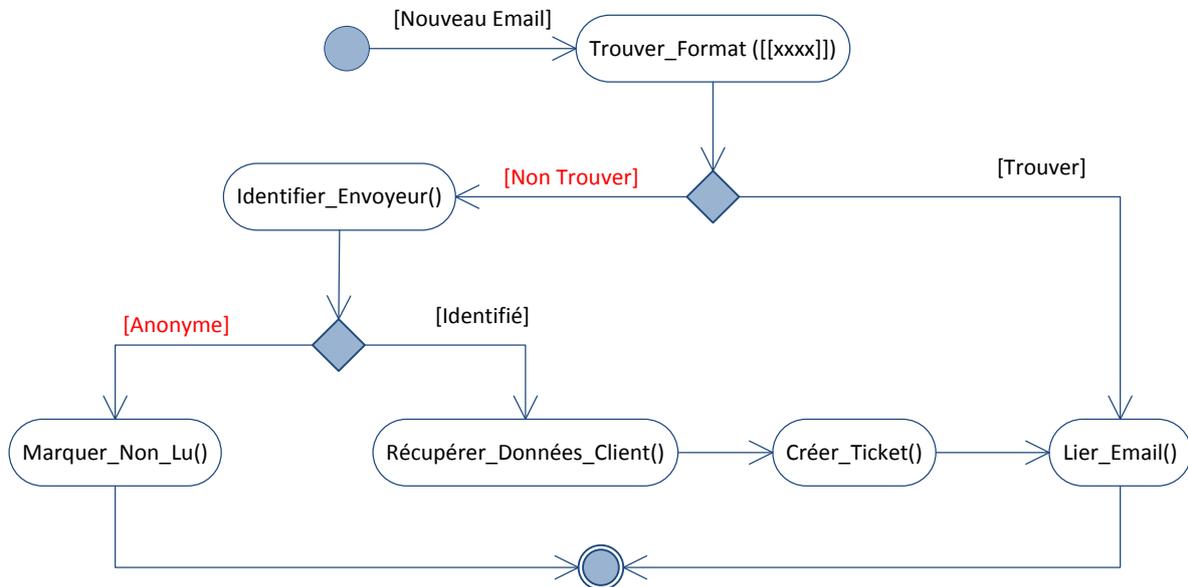


Figure 20 : Diagramme d'activités du traitement d'un email récupéré par B-WARN.

F. Service de notification

Le service de notification est un module complémentaire dont tout le système est pleinement indépendant, mais il offre à B-WARN une fonctionnalité surprenante qui consiste à notifier les utilisateurs de toutes modifications de leur portefeuille de service, comme il pourrait être utilisé pour contrôler la gestion soit des tickets des clients importants à l'entreprise, soit des projets importants.

Plusieurs notifications prédéfinies, gérées par ce module, sont regroupées en deux types :

- Les notifications provoquées par les événements du système :
 - o Un nouveau ticket assigné.
 - o La demande d'escalade acceptée ou refusée.
 - o Un email est récupéré, et lié à un ticket persistant.

Les messages de notifications engendrés par ces événements sont envoyés respectivement à l'employé auquel le ticket est assigné, au créateur de la demande d'escalade, et au responsable du ticket.

- Les notifications conditionnelles, lancées quand leurs conditions sont valides :
 - o Le nombre des tickets a atteint une limite fixée par l'administrateur.
 - o Le nombre des demandes d'escalade a atteint une limite fixée par l'administrateur.
 - o La création d'un nouveau ticket.

Les conditions de ces notifications peuvent comprendre plusieurs critères : le projet, le client, la catégorie du ticket et la priorité. Le ou les destinataires, qui sont les utilisateurs locaux de B-WARN peuvent différenciés d'une instance à une autre selon les besoins.

Pour chacune des notifications du système, plusieurs instances pourront être créées en associant différentes conditions à chacune (pour les notifications conditionnelles). De même, le message de notification pourra être modifié dans les différentes instances de la même notification.

La figure qui suit (Figure 21) représente une instance de notification conditionnelle nommée *Iris 7.1 – High – Ticket Created*.

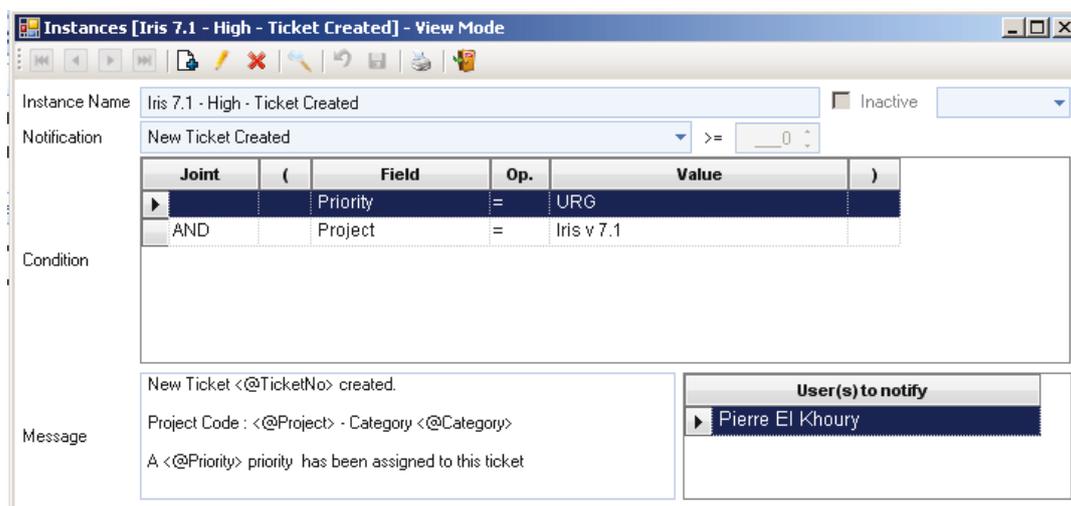


Figure 21 : Instance de notification conditionnelle de B-WARN.

L'instance affichée dans la figure précédente sera déclenchée en créant un ticket du projet *Iris v 7.1*, auquel une priorité *URG* est associée.

Le message de notification de cette instance pourra être changé en gardant les chaînes de caractères de la forme <@_ _ _> qui seront remplacés, lors de la validation de l'instance, par

les données convenables du ticket créé. Par la suite, le message sera envoyé aux employés spécifiés dans la table nommée *User(s) to notify*.

Le processus de validation des instances de notifications des utilisateurs est résumé dans le diagramme de séquences de la Figure 22.

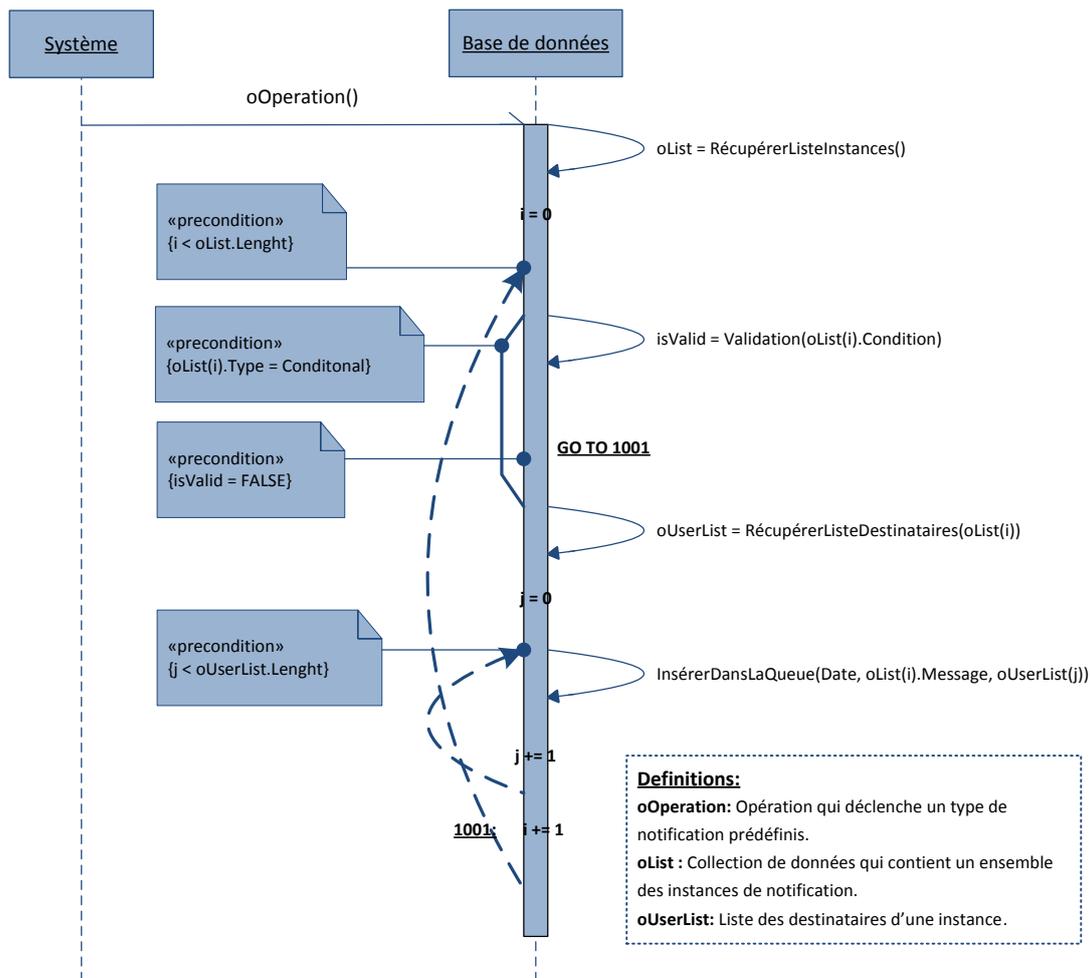


Figure 22 : Diagramme de séquences du processus de validation des instances de notification dans B-WARN.

Après validation de leurs conditions ou occurrence de leurs événements, toutes les instances des notifications génèrent des messages de notifications à leurs destinataires. Ces messages sont sauvegardés dans une file avec la date de création et l'identité du destinataire.

La file des messages de notifications peuvent être consultée manuellement par chaque utilisateur, en appuyant sur le bouton *Mes Notifications* de la barre d'outils de la Zone 1 de l'interface principale. Par ailleurs, le système peut informer les utilisateurs de la présence de

quelques messages dans la file au moyen d'un avertissement qui s'affiche sur leurs écrans. L'utilisation de cette dernière fonctionnalité est optionnelle ; elle sera fixée par les paramètres généraux du système.

Une fois un message de notifications est aperçu par l'utilisateur, celui-ci pourra :

- Marquer le message lu, ainsi le message sera-t-il effacé de la file.
- Demander au système de le notifier plus tard, et le message réside dans la queue.
- Ignorer le message. Le message sera donc effacé de la queue.

Pour offrir une meilleure performance, ce module contient un historique de tous les messages de notifications, le temps exact de notification et la réponse sélectionnée par chaque destinataire.

G. Statistiques

Ce module est responsable d'exploiter la base de données du système et de transformer les données en des graphes statistiques qui seront utilisés par la direction de l'entreprise comme base de toutes nouvelles stratégies financières et managériale de l'entreprise, spécifiquement celles reliées au service client.

Ce module est seulement composé de deux interfaces graphiques. L'interface initiale du module nommée *Charts Manager* ou *Gestionnaire des Cartes*, sert à créer différents cartes graphiques qui visualisent plusieurs ensembles de données.

Plusieurs types de cartes graphiques 2 et 3 dimensionnelles sont offertes par cette interface: les histogrammes, les cartes chronologiques, les cartes des colonnes et cylindres clustérisées et plusieurs autres modèles.

Toutes les cartes graphiques créées sont caractérisées par leurs créateurs à l'aide de plusieurs paramètres. Ces paramètres sont regroupés en deux catégories :

- Les paramètres graphiques qui servent à modifier les propriétés d'affichage des cartes : le type, la couleur, les annotations, la légende etc.
- Les paramètres de données qui filtrent les données récupérés et affichées par les cartes.

Trois paramètres nécessaires devraient être spécifiés pour chacune des cartes créées:

- L'axe X, qui représente les données affichées horizontalement par la carte.
- L'axe Y, qui symbolise les données affichées en verticale.
- L'indicateur qui définit la variable de mesure de la carte.

Par la combinaison des plusieurs valeurs possibles des ces paramètres, une diversité de cartes pourrait être créée par les utilisateurs du B-WARN affichant des ensembles de données avantageux à l'entreprise, comme:

- Carte 1 : Affichage du nombre de tickets assignés à chaque utilisateur pour chaque projet.

Axe X = Utilisateur ; Axe Y = Projet ; Indicateur = Nombre des tickets

- Carte 2 : Le temps de service (en min) dépensé pour chaque client par chaque utilisateur.

Axe X = Client ; Axe Y = Utilisateur ; Indicateur = Minute de service

La figure ci-dessous (Figure 23) représente la fenêtre de saisie *Charts Manager*. Elle montre en haut les critères de données, à gauche les paramètres graphiques de la carte, et la carte générée au centre.

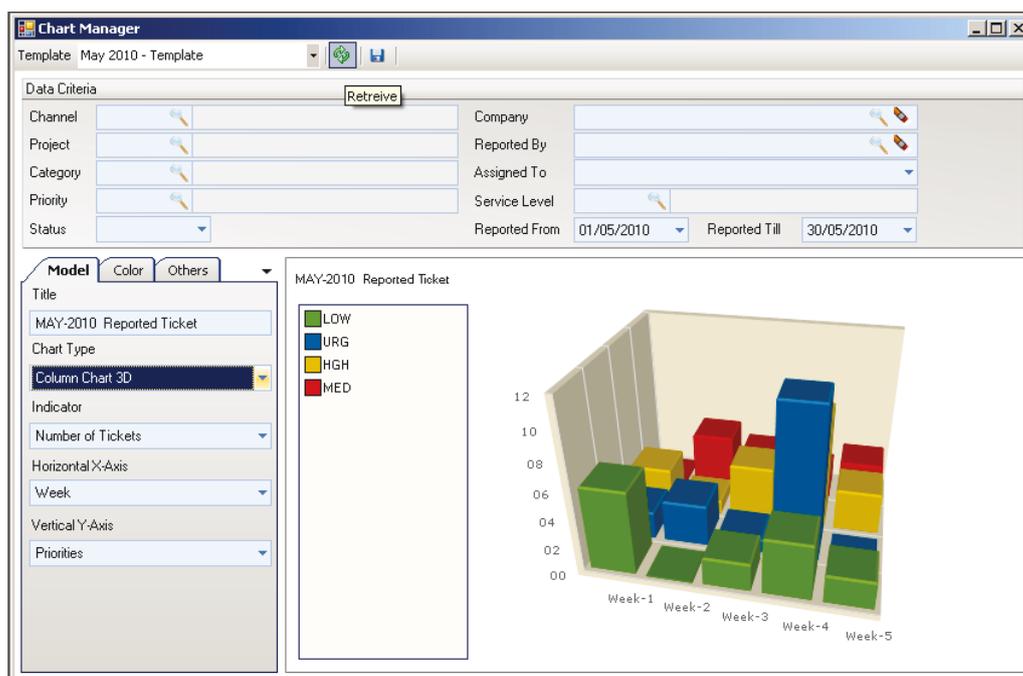


Figure 23 : Fenêtre de saisie *Charts Manager* de B-WARN.

Les cartes graphiques créées, et tous leurs paramètres sélectionnés, sont sauvegardés dans le système. Elles seront affichées dans le menu de ce module exposé dans l'interface principale de l'application.

Une fois une carte est visitée par un utilisateur du système, une deuxième interface de ce module s'ouvre en affichant la carte sauvegardée. Pour limiter la visualisation des cartes aux utilisateurs non autorisés, les permissions d'affichage du module de sécurité seront appliquées à chacune des cartes sauvegardées comme étant une interface indépendante.

H. Rapports

B-WARN offre à ces utilisateurs plusieurs rapports pour afficher toutes les données sauvegardées par le système. La liste des rapports sera affichée et filtrée par les droits d'exécution de chaque utilisateur.

En vue d'offrir une meilleure utilisation des résultats affichés, chaque rapport contient plusieurs critères de sélection des données et une fonction de regroupement à deux niveaux des données affichées.

Tous les rapports de ce module peuvent être imprimés ou transformés en des fichiers textes, Excel ou PDF, selon les besoins des utilisateurs.

2.2 LIVE B-WARN

LIVE B-WARN, l'application Web du système qui est considérée comme un composant avantageux, a été développée pour être la porte d'interaction des clients avec B-WARN. Elle offre à l'entreprise une meilleure relation avec ces clients ; une relation claire et dynamique.

Cette application offre aux clients la possibilité de superviser la progression de leurs tickets, de les modifier avec quelques fonctions de manipulation limitées, et permet aux utilisateurs de créer des tickets neufs d'une façon simple et pratique.

Les utilisateurs de LIVE-B-WARN sont les clients de l'entreprise, ils accèdent à l'application via internet, en saisissant dans le navigateur web l'adresse du serveur sur lequel elle est publiée. Ces utilisateurs enregistrés dans B-WARN sous forme des contacts seront

autorisés d'utiliser LIVE B-WARN en leur accordant un mot d'utilisation et un mot de passe. Les accès des utilisateurs pourront être limités par des permissions relatives à chacune des fonctions de l'application. De même, les données affichées aux utilisateurs pourront être soit ouvertes pour englober tous les tickets de leurs compagnies soit limitées aux tickets reportés par chacun d'eux.

Après l'authentification de l'utilisateur, toutes les informations de sa compagnie seront chargés par la session web (le nom, le projet, la priorité etc.) et il sera piloté vers la page d'accueil de LIVE B-WARN. Cette page contient le menu des différentes fonctions de l'application, un texte descriptif de l'application et un autre qui affiche les coordonnées de l'entreprise spécifiées dans les paramètres généraux de B-WARN.

Outre les fonctions de manipulation des tickets, deux outils de management sont offerts par LIVE B-WARN:

- Un rapport qui liste les tickets du client. Ce rapport pourra être imprimé ou sauvegardé sur le disque dur de la station de l'utilisateur en forme de fichier texte, fichier Excel, fichier PDF et plusieurs autres formats.
- Une carte graphique similaire aux cartes du module de statistique de B-WARN. Cette carte pourra être paramétrée selon le choix de l'utilisateur.

De même, les informations personnelles des utilisateurs, sauvegardées par le système, seront affichées à chacun d'eux. Ces données pourront être modifiées et mises à jour, en utilisant quelques fonctions de manipulation limitées :

- Ajout d'une adresse et changement de l'adresse utilisée par défaut.
- Ajout d'un numéro de téléphone et changement du numéro utilisé par défaut.
- Ajout d'une adresse email et changement de celle utilisée par défaut.

En outre, comme la plupart des applications Web, LIVE B-WARN offre aux utilisateurs la possibilité de changer leur mot de passe pour des raisons de sécurité.

Le diagramme de cas d'utilisation de la Figure 24 résume toutes les fonctions intégrées dans LIVE B-WARN.

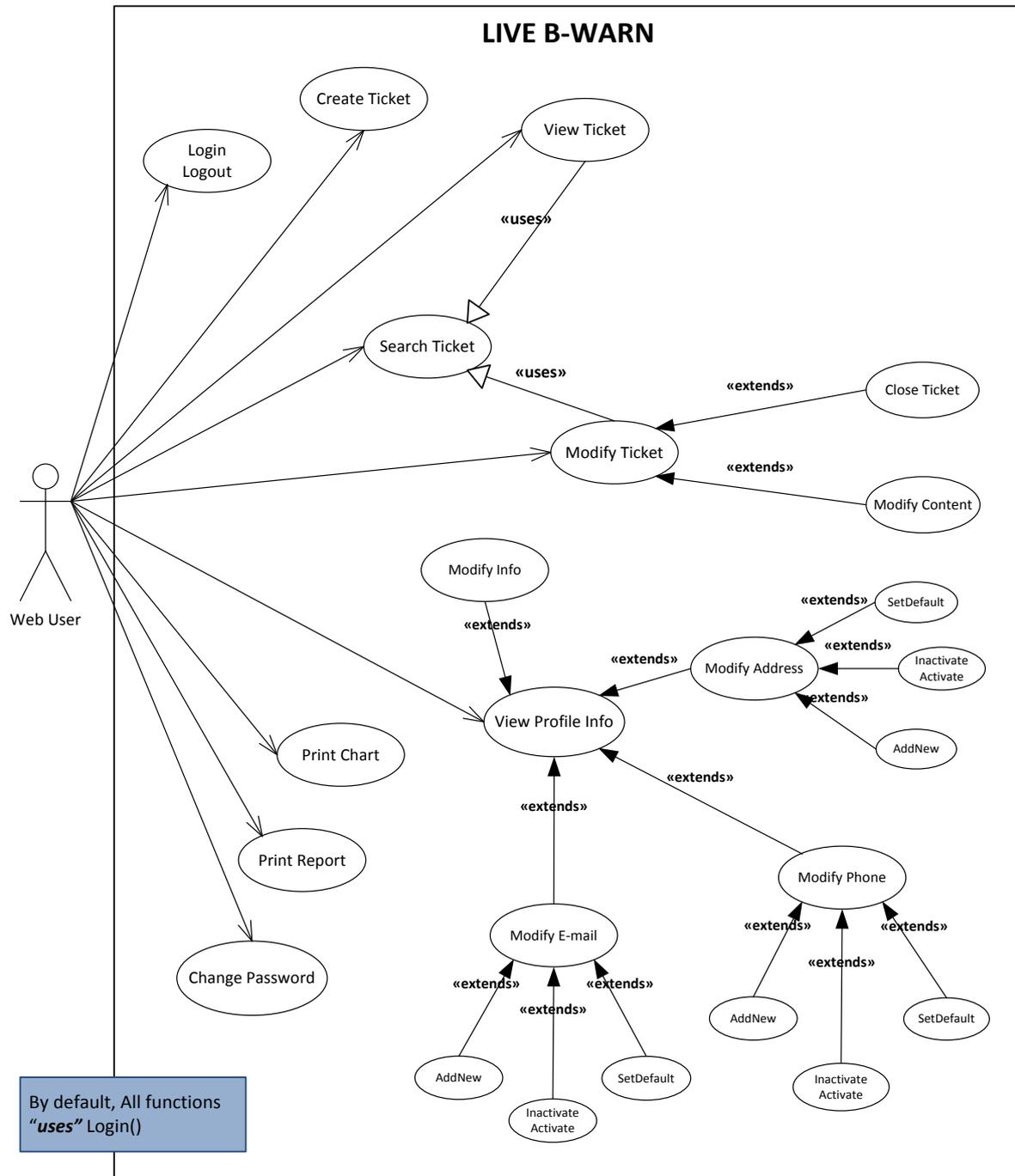


Figure 24 : Diagramme de cas d'utilisation d'LIVE B-WARN.

Finalement, les données manipulées par LIVE B-WARN sont récupérées et sauvegardées dans la même base de données que B-WARN. Cela permet une mise à jour instantanée des données, et par la suite offre aux employés la possibilité de traiter immédiatement les nouveaux tickets créés par les clients.

3. Interprétation technique

Les deux applications B-WARN et LIVE B-WARN sont développées par des technologies .NET Framework 3.5. Elles sont conçues pour opérer sous les différentes versions du système d'exploitation Microsoft Windows (XP, Vista, Windows 7, etc.)

Dans ce qui suit, une description technique détaillée de chacun des composants du projet.

3.1 La base de données

La base de données, centralisée pour les deux applications du système, est créée en SQL Server en utilisant l'outil SQL Server Management Studio 2008 (SSMS) pour faciliter le développement.

Cette base de données est relationnelle. Elle est formée de 60 tables données, reliées par plusieurs relations de sorte de garantir fortement l'intégrité des données qui y sont stockées. Sa taille est approximativement 44 Mo, avec des tables vides de toutes données à l'exception des valeurs et types prédéfinies.

Tous les noms tables de la base de données commencent par un préfixe qui sert à les regrouper et les ordonner visuellement. Ce préfixe représente le module du système auquel appartiennent les tables, par exemple : « TK_ » pour les tables des données du module *Gestionnaire des Tickets*, « SYS_ » pour les tables du module *Paramètres Généraux* etc.

Sept diagrammes, qui représentent les sept modules du système, sont créés pour faciliter la compréhension de la structure de la base de données. Ces diagrammes visualisent les différentes tables et relations du module.

Les différents types et données prédéfinies du système sont stockés dans une seule table nommée *SYS_Types*. Elle contient les trois colonnes : *TypeID*, *TypeDesc*, *DataTable*. La première colonne est utilisée pour lier les types aux autres tables de données, la deuxième pour l'affichage des données et la troisième pour définir le type de données par exemple :

[TypeID = 21] [TypeDesc = New] [DataTable = TicketsStatuses]

En plus, plusieurs vues sont reproduites afin de regrouper les tables fréquemment utilisées et éviter par la suite de récrire des requêtes complexe à chaque besoin des données.

De même, les fonctions et les procédures stockées ont été utilisés pour faciliter la récupération des données compliquées, comme l'état actuel d'un ticket ou l'utilisateur auquel le ticket est actuellement assigné, qui nécessitent une lecture de plusieurs tables de données et une organisation spécifiques des lignes de données. Par exemple, une des fonctions stockées nommée `fnGetTicketAssignedTo(int, date)` prend en paramètre l'identité d'un ticket et une date, recherche dans l'historique du ticket, et renvoi le nom de l'employé auquel ce ticket est assigné dans la date spécifié (voir l'Annexe 3).

Les déclencheurs (Triggers) ont été notamment utilisés dans les tables de données reliées au module *Service de Notification*. Ce module qui effectue la plupart de ces fonctions au niveau de la base de données (voir Figure 22 : Diagramme de séquences du processus de validation des instances de notification dans B-WARN.) est formé d'un ensemble de déclencheurs qui servent à valider les conditions des instances des notifications à chaque événement d'insertion ou de mise à jour, et insérer par la suite les messages dans la file.

En outre, les déclencheurs sont utilisés pour générer les numéros de série des tickets, et pour créer les états initiaux des tickets après leurs créations (voir l'Annexe 4).

Enfin, plusieurs indexes sont créés dans différentes tables de données afin de les ordonner par les champs de données fréquemment utilisées dans les requêtes de recherche SQL, comme les champs `TicketNo` et `ReportedOn` de la table `TK_Tickets`. Ces indexes permettent une extraction rapide des données, ce qui aide à optimiser le temps d'exécution des requêtes complexes.

3.2 B-WARN

L'application bureautique B-WARN est développée en utilisant l'outil Microsoft Visual Studio 2008 pour faciliter le développement des solutions, assister par Microsoft Visual SourceSafe pour gérer les accès des développeurs aux différents fichiers des solutions, et pour protéger l'équipe de développement de toute perte de fichier accidentelle.

Une librairie de contrôles graphiques est essentiellement utilisée pour enrichir les interfaces graphiques de cette application. Cette librairie nommée *Net Advantage for .NET volume 3 CRL 2.0* développée par *Infragistics*, comporte des composants graphiques qui aident à

améliorer la performance de l'application, en modernisant l'interaction de ces interfaces avec les utilisateurs.

B-WARN est une application orientée objet, formée de plusieurs projets de types bibliothèques de classes séparés physiquement, qui génèrent chacun après compilation des assemblages (Assemblies) DLLs et qui seront référencés par les autres projets, selon les besoins. Un seul projet responsable du lancement de l'application génère un fichier EXE.

Trois projets spécifiques forment la librairie interne du système :

- **Shared**, qui contient plusieurs fonctions fréquemment utilisées comme `GetDataSet()` et `GetDataReader()`, qui sont des fonctions de récupérations de données.
- **Shared.WinControls** et **Shared.UserControls**, qui contient des composants graphiques paramétrés pour l'utilisation dans les interfaces de B-WARN.

Les modules du système décrits précédemment, sont développés selon une méthode unifiée, considérée un standard de développement à PIXEL. Cette méthode consiste à subdiviser chaque module en cinq projets de types classes bibliothèques nommés : `DataSets`, `DataAccess`, `BusinessRules`, `UI` et `Reports`. Les cinq projets de chaque module sont placés sous le même répertoire et sont regroupés sous le même espace de nom (namespace) qui représente le nom du module. Dans ce qui suit une description de ces projets et de leurs composants essentiels:

1. [Module].DataSets

Ces projets comportent des classes *DataSet*, objets ADO.NET. Ces classes représentent un cache en mémoire des données. Ils sont constitués d'une collection d'objets *DataTable* qui représentent les tables de la base de données, et liés entre eux par des objets *DataRelation*. Ces classes respectent l'intégrité des données à l'aide des objets *UniqueConstraint*, *ForeignKeyConstraint* et plusieurs autres contraintes.

2. [Module].DataAccess

Ces projets comportent des classes qui contiennent des composants *SqlDataAdapter*. Ces composants représentent chacun un ensemble de commandes de données et une connexion à la base de données, qui permettent de remplir les *DataSets* du projet précédent du même module et de mettre à jour la base de données SQL Server.

Chacune des classes de ces projets contient trois propriétés publiques :

```
Public WriteOnly Property ConnectionString() As String
Public ReadOnly Property ErrorOccured() As Boolean
Public ReadOnly Property ErrorMessage() As String
```

Ces propriétés servent respectivement à initialiser les paramètres de connexion de la base de données utilisée par les objets *SqlDataAdapter*, à indiquer l'occurrence d'une erreur pendant l'exécution, et à récupérer le message de l'erreur (voir Annexe 5).

3. [Module].BusinessRules

Ces projets utilisent (ou référencent) les deux projets définis ci-dessus. Ils contiennent plusieurs fonctions qui utilisent des instances des classes du projet *[Module].DataAccess* pour remplir ou mettre à jour les données qui existent dans les instances des classes du projet *[Module].DataSets*. Les classes de ce projet possèdent deux propriétés publiques :

```
Public ReadOnly Property ErrorOccured() As Boolean
Public ReadOnly Property ErrorMessage() As String
```

Ces propriétés renferment les mêmes valeurs des propriétés de la classe *DataAccess* respective. En plus, ces classes contiennent deux fonctions additionnelles:

```
Public Sub SetDefaultValues(...)
Private Function DataIsValid(...) As Boolean
```

La première fonction prend en paramètre un objet *DataSet* et initialise les champs de données de ces tables. La deuxième, est privée pour la même classe, elle sert à valider les données avant d'appeler la fonction de mise à jour (voir Annexe 6).

4. [Module].UI

Ces projets contiennent les interfaces graphiques responsables des communications et des interactions avec les utilisateurs. Les différents composants de ces interfaces sont liés aux sources de données (classes du projet *[Module].DataSet*) en utilisant la classe *DataBinding()*, afin d'afficher les données d'une manière lisible par les utilisateurs. Ces données sont mises à jour en appelant les fonctions des classes du projet *[Module].BusinessRules*.

5. [Module].Reports

Ces projets sont totalement indépendants des quatre projets précédents. Ils contiennent tous les rapports du module. Ces rapports sont développés en utilisant Crystal Reports en forme de classes RPT.

Tous ces projets des différents modules sont regroupés dans B-WARN sous le module nommé *Rapports*. Ce module qui n'existe pas physiquement est un simple regroupement visuel de tous les projets *[Module].Reports*.

Les modifications graphiques des propriétés de l'interface principale de B-WARN (Dispositions des fenêtres, affichage du menu, etc.) réalisées par les utilisateurs, sont sauvegardées localement sur chaque station par deux fichiers surnommés *UI_DockPanels.dat* et *UI_Toolbar.dat*. Ils sont responsables de sauvegarder ces modifications afin d'initialiser l'interface principale à chaque démarrage de B-WARN.

Enfin, un composant *Windows Service* est développé séparément des autres projets. Ce composant appelé *B-Warn.Extension.exe* est considéré comme une partie détachée du module *Gestionnaire des Emails*. Il est responsable de consulter les comptes emails, sauvegardés dans le système avec le paramètre *Auto-synchroniser = VRAI*, à un intervalle de temps donné, et de récupérer tous les nouveaux messages de leurs boîtes de réception. Ce composant est installé, indépendamment de l'application, sur une station du réseau de l'entreprise et lancé automatiquement avec le démarrage du système d'exploitation Windows.

Le diagramme des composants de la Figure 25 illustre la structure physique de B-WARN en utilisant un seul module *Gestionnaire des Tickets* (pour simplifié le diagramme), et expose les relations de référence entre les différents fichiers composants l'application.

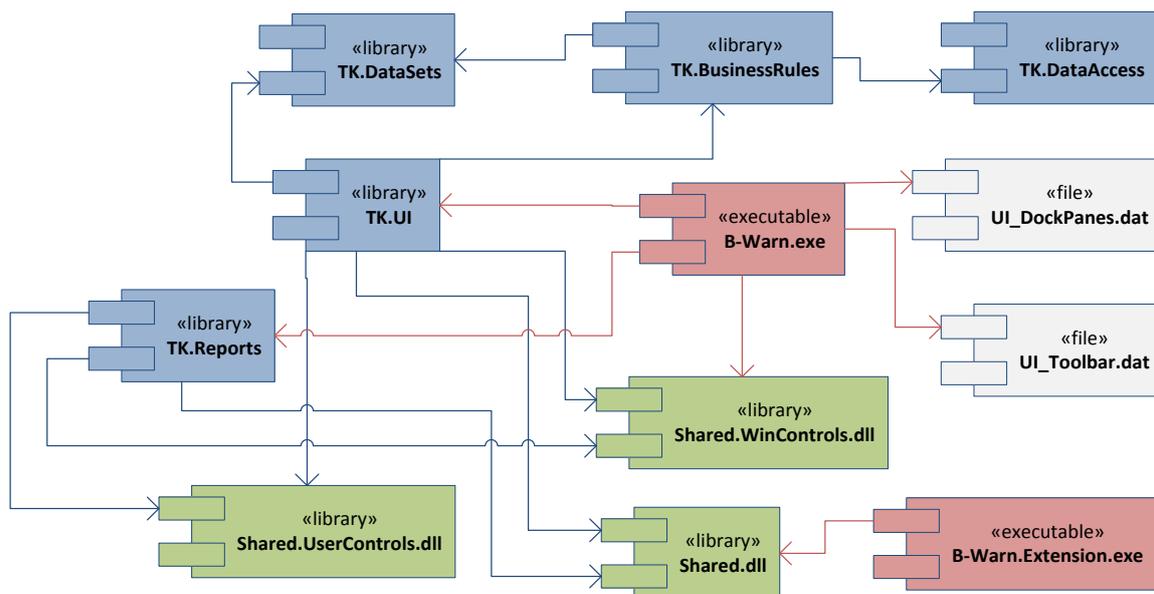


Figure 25 : Diagramme de composants de la structure physiques de B-WARN.

3.3 LIVE B-WARN

LIVE B-WARN, l'application Web du système, est développée en ASP.Net en utilisant les mêmes outils de développement Microsoft Visual Studio 2008 et Microsoft Visual SourceSafe.

Cette application est de type *AjaxEnabled* qui permet aux développeurs d'implémenter les fonctionnalités de la technologie AJAX. Cette technologie permet la mise à jour des données de quelques parties des pages web, sans rechargement complet des pages du navigateur. Elle facilite l'interaction entre les utilisateurs et les pages de l'application et minimise le temps d'attente des utilisateurs.

Quelques composants graphiques web offerts par *Net Advantage for .NET volume 3 CRL 2.0* sont utilisés pour enrichir les pages de l'application. Ces composants qui possèdent plusieurs événements du côté client (client-side), et qui sont aussi typés *AjaxEnabled*, ont permis aux développeurs de créer un grand nombre des fonctions en JavaScript qui s'exécute au côté client, ce qui a réduit les allées et retours vers le serveur. Mais l'utilisation des ces composants est limitée par les besoins, et cela pour éviter d'alourdir les pages.

Le déploiement de cette application est basé sur trois fichiers essentiels:

- **Web.config** : Fichier essentiels des applications ASP.Net. Ce fichier de format XML, décrit le système de configuration de l'application, son langage de développement et les paramètres de compilation. En plus, il contient des paramètres spécifiques de l'application comme le *ConnectionString* qui représente la chaîne de connexion à la base de données.
- **Global.asax** : Il contient des lignes de code pour répondre aux événements au niveau de l'application ou des sessions déclenchés par ASP.NET. Plusieurs événements sont gérés par ce fichier, tels que *Application_Start()*, qui permet d'initialiser quelques variables globales de l'application, et *Application_Error()*, qui définit le comportement de l'application en cas d'exceptions non gérées.
- **Licenses.licx** : C'est un fichier traditionnel pour ce type d'applications. Il est créé et modifié automatiquement par Visual Studio. Il contient toutes les informations des licences des composants utilisés dans l'application.

Les *Web Services* sont fréquemment utilisés dans cette application pour pouvoir utiliser les fonctionnalités AJAX des composants graphiques de *Net Advantage*, et aussi pour pouvoir minimiser la création des fonctions du côté serveur (server-side) qui nécessitent le rechargement des pages après chaque appel.

Les différentes pages de cette application sont formées de deux fichiers *Page.aspx* et *Page.aspx.vb*. Le premier fichier en format HTML contient les composants graphiques et toutes les fonctions qui s'exécutent du côté client, écrites en JavaScript, tandis que le deuxième contient les codes qui s'exécutent du côté serveur en forme de classe VB. De même, la plupart des pages de l'application contiennent le composant *ScriptManager* qui sert à enregistrer les *Web Services*, pouvoir par la suite les appeler du côté client.

4. Implémentation

Le déploiement de B-WARN SUITE nécessite un simple réseau LAN qui comporte deux serveurs : un serveur de données et un serveur web.

Sur le premier serveur, la base de données centralisée de B-WARN SUITE sera placée. Ce serveur devrait être accessible par les différentes stations du réseau afin de permettre aux applications du système de se connecter à la base de données.

Sur le deuxième serveur, LIVE B-WARN, l'application web, sera publiée. Ce serveur possédant un IP publique, sera plus tard accéder par les stations externes des utilisateurs distants cette application en saisissant l'IP et le nom de l'application dans leurs navigateurs web connectés à travers le réseau Internet.

B-WARN.extension.exe, le *Windows Service*, considéré comme une partie détachée de B-WARN, sera enregistré sur l'une des stations du réseau, une station connectée à l'internet. Comme ce service sera lancé en démarrant le système d'exploitation, il est recommandé de l'enregistrer sur le serveur web qui sera rarement éteint.

B-WARN, l'application bureautique du système, sera installée sur chacune des stations du réseau de l'entreprise.

Le réseau nécessaire et les installations des différents composants de B-WARN SUITE sont schématisé par la figure suivante:

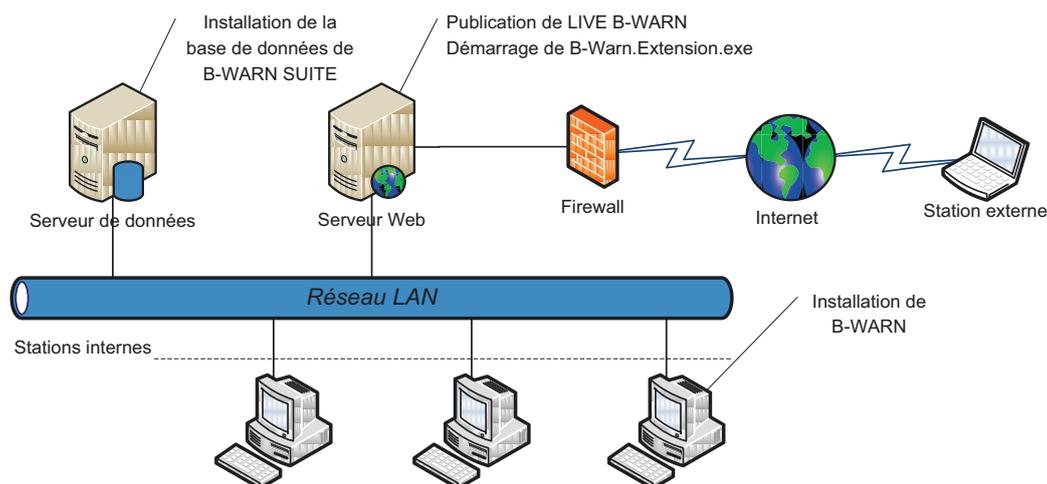


Figure 26 : Réseau et installations nécessaires pour le déploiement de B-WARN SUITE.

Les configurations matérielles requises de chacune des machines du réseau sont déduites d'après les logiciels requises au déploiement du système, en addition à l'espace disque nécessaires à l'application, et la mémoire utilisée durant son exécution. Toutes ces configurations sont résumées par le tableau qui suit:

Table IX : Configuration des machines requise pour le déploiement de B-WARN SUITE.

Composant	Configuration requise
Serveur de données	
Système d'exploitation	Microsoft Windows server 2003 SP1, SP2 ou une version ultérieure.
Autres logiciels	Microsoft SQL Server 2008 R2 Enterprise (32 bits ou 64 bits)
Processeur	Type de processeur : - Processeur compatible Pentium III ou supérieur Vitesse du processeur : - Minimum : 1,0 GHz - Recommandé : 2,0 GHz ou plus
Mémoire	RAM : - Minimum : 1 Go - Recommandé : 4 Go ou plus
Espace disque	Moteur de la base de données : 711 Mo Base de données (vide): 44 Mo
Serveur Web	
Système d'exploitation	Microsoft Windows server 2003 SP1, SP2 ou une version ultérieure.
Autres logiciels	Microsoft .NET Framework 3.0 ou plus récent. Microsoft Internet Information Server (IIS) 5.1 ou plus récent.
Processeur	<i>Les configurations sont modifiées selon le nombre des utilisateurs concurrents.</i> Type de processeur : - Processeur compatible Pentium III ou supérieur

Composant	Configuration requise
Mémoire	Vitesse du processeur : - Minimum : 1,0 GHz - Recommandé : 2,0 GHz ou plus
	RAM : - Minimum : 1 Go - Recommandé : 4 Go ou plus
Espace disque	12 Mo
Stations internes (Utilisateurs locaux)	
Système d'exploitation	Microsoft Windows XP Édition familiale ou Microsoft Windows XP Professionnel, avec SP 2 ou une version ultérieure.
Autres logiciels	Microsoft .NET Framework 3.0
Processeur	Pentium 1 GHz ou supérieur.
Mémoire	512 Mo ou plus
Espace disque	11.5 Mo
Stations externes (Utilisateurs web)	
Système d'exploitation	-
Autres logiciels	Microsoft Internet Explorer 6 ou une version ultérieure. Firefox, Mozilla (version compatible avec .NET framework 3.0). Les navigateurs web doivent permettre l'exécution des JavaScript.
Processeur, Mémoire, Espace disque	-

L'installation de B-WARN sur les stations internes du réseau pourrait être réalisée de deux manières. La première consiste à installer directement l'application sur chaque station à l'aide d'un assistant d'installation *B-WARN_Installer.exe*, tandis que la deuxième utilise la technologie de déploiement *ClickOnce* de Microsoft. Cette méthode consiste à publier les différents fichiers de l'application sur un des serveurs, en utilisant le projet *B-WARN.Publisher* qui crée une page web spéciale pour l'application dans le répertoire IIS (Internet Information Services) du serveur. Les stations du réseau accédant à cette page en saisissant l'adresse convenable dans navigateur web. Par la suite, par un simple clic de souris l'installation et le lancement de l'application seront réalisées.

La deuxième méthode est plus recommandée ; elle permet d'installer et de mettre à jour automatiquement l'application avec un minimum d'intervention de l'utilisateur. En effet, celle-ci vérifie à chaque démarrage de l'application la disponibilité d'une nouvelle version et propose à l'utilisateur de réaliser la mise à jour. Cette méthode nécessite l'installation de Microsoft Visual Studio 2008 sur le serveur web pour pouvoir ouvrir *B-WARN.Publisher* afin de publier les versions de l'application.

Chapitre IX : Conduite du projet

Ce chapitre décrit les problèmes majeurs, techniques et organisationnels, rencontrés durant le développement du projet. Cette description comportera la définition des problèmes et leurs conséquences sur le déroulement du travail. De même, le plan de développement appliqué sera décrit, en mettant en lumière les délais et les retards des différentes tâches.

A la fin, une évaluation personnelle sera décrite en critiquant tous les points faibles et forts déduits, grâce à la supervision des différentes phases du développement et surtout des façons de traitement des obstacles rencontrés.

1. Problèmes rencontrés

Comme dans tout cycle de développement d'un logiciel, plusieurs problèmes furent rencontrés. Certains problèmes, étant des fautes de conduite du projet ou des agents externes inattendus et incontrôlés, ont touché la progression du développement du projet ; d'autres avaient une influence partielle sur l'une de ses parties.

En effet, B-WARN SUITE, étant un produit interne de l'entreprise et du point de vue management, n'a pas de rendement financier direct sur celle-ci. Il était soumis à plusieurs menaces qui ont influencé le déroulement du travail et surtout la répartition des tâches et la considération des dates des intégrations des composants spécifiés par la planification initiale des tâches.

Ces problèmes sont regroupés en trois catégories : problèmes d'ordre organisationnels, problèmes de conception et problèmes techniques. Dans ce qui suit, une description détaillée des problèmes majeurs rencontrés dans chaque catégorie, et leurs conséquences sur le déroulement du travail.

1.1 Problèmes d'ordre organisationnels

L'équipe du développement de B-WARN SUITE était soumise à plusieurs changements et modifications structurels, ce qui a eu un impact direct sur le déroulement du développement des différentes tâches, et sur le respect du délai prévu pour chacune d'elles. Trois problèmes majeurs ont été distingués. Ils se résument par ce qui suit.

1.1.1 Charges et responsabilités externes

Dès le début du projet, c'était bien connue que les quatre membres de l'équipe de développement ne sont pas entièrement voués à ce projet, par contre ils étaient responsables en même temps d'autres tâches externes, notamment celles de service client. Ainsi, l'estimation des heures de travail par jour pour chaque personne a été de 5 heures consacrées au projet sur les 8 heures de travail quotidien.

Au cours du développement du projet, le nombre de tâches attribuées aux développeurs a accru d'une manière inattendue. Cet accroissement des tâches était dû aux améliorations et aux nouveaux projets des clients, sans oublier les requêtes de services nécessaires pour le service client. Cette situation ambiguë de répartition des tâches, a notamment minimisé les heures de travail consacrées au projet pour atteindre entre 3 et 4 heures par jour, et même parfois moins.

Cette diminution des heures de travail consacrées au développement du projet avait son impact sur la plupart des tâches, surtout celles du codage des composants. Mais les efforts exceptionnels fournis par les membres de l'équipe, et les heures de travail supplémentaires, ont aidé à rattraper certain temps perdu. Cependant, cela n'a pas été suffisant pour respecter les dates limites de toutes les tâches.

Avant le début du projet, deux mises en place d'IRIS ont été achevées, et les requêtes intenses des nouveaux clients sont toujours attendues après cette phase. C'est pourquoi, ce problème était prévu dès le début du projet, mais la planification initiale du projet a été si optimiste que cette situation n'a pas été sérieusement considérée.

1.1.2 Restructuration du personnel

Au début du développement du projet, comme c'est déjà mentionné, j'ai été le responsable du service client. Un mois après le début du projet, à cause de l'émigration inattendue du membre le plus âgé de l'équipe de développement de PIXEL, une modification structurelle a été effectuée. Elle a été conçue pour répartir les responsabilités tenues par ce membre sur les autres, spécialement sur ceux les plus chevronnés.

La nouvelle structure m'a libéré de mes responsabilités de responsable du service client, et m'a chargé d'une implémentation déjà commencée d'une version d'IRIS à l'étranger. Cette nouvelle fonction a exigé de ma part un voyage mensuel de presque quinze jours.

Malgré mes nouvelles responsabilités, j'ai toujours pris en charge le développement de B-WARN SUITE. Mais mes absences avaient leur influence négative sur les communications avec l'équipe du développement, ce qui a ralenti le déroulement du travail. De même, ces absences ont exigé de retarder quelques tâches, surtout celles de conception.

Ce problème n'était attendu ni par l'équipe du projet ni par l'administration de l'entreprise. La seule solution trouvée a été de réorganiser les tâches avant chaque voyage afin de minimiser la perte du temps.

1.1.3 Changement des priorités

Au début de ce projet, l'administration de PIXEL lui a attribué une grande priorité, étant donnée son importance sur l'organisation de l'entreprise et sa progression. Mais au fur et à mesure cette priorité a diminué à cause des agents externes variés inattendus : la diminution du personnel, l'accroissement des améliorations commandées par les clients...

Le changement de priorité a débuté deux mois après le commencement du développement. Cela s'est traduit par la diminution du nombre des développeurs pour aboutir, au début du quatrième mois, à la prise de la décision d'ajourner le projet. Cette décision a été une obligation puisque la réponse aux besoins des clients était plus importante que l'achèvement du projet.

De mon côté, ce projet a été de la même importance que la réponse aux besoins des clients, et parfois plus important que les besoins non pressants qui peuvent être ajournés, et qui sont pour l'entreprise une source de rendement financier qui peut être retardé sans toucher à la satisfaction des clients.

Vu les différentes opinions qui ont accompagné la prise de cette décision, j'ai proposé de continuer tout seul le développement du projet, à condition que les autres membres de l'équipe assistent au développement selon leurs disponibilités. Ma proposition fut acceptée par l'administration de l'entreprise et toutes les ressources matérielles (équipements, bibliothèques, outils...) ont été mises à ma disposition, selon mes besoins.

La modification de priorité du projet, et ma décision de continuer le reste du développement ont affecté la progression du développement des tâches, surtout de celles qui devraient être développées en parallèle. Ce retard a été prévu dès la prise de la décision mais, il valait mieux que d'arrêter le développement, avec une remise en marche planifiée.

1.2 Problèmes de conception

Les problèmes de conception avaient de même une influence remarquable sur le développement de B-WARN SUITE. Leurs impacts ont été mineurs par rapport aux problèmes organisationnels, mais ils ont influencé d'une façon ou d'autre le déroulement du travail. Ces problèmes ont été, dans leur majorité, des fautes de conception et des cas d'utilisation non considérés par l'analyse des différents modules du projet. Dans ce qui suit une description des cinq problèmes majeurs techniques rencontrés.

1.2.1 Utilisation de SQL Server 2008 à la place de SQL Server 2005

Au début la conception du projet, SQL Server 2005 a été l'SGBD choisi pour le développement de la base de données du système. De même, la plate-forme *SQL Server 2005 Notifications Services* a été la base du module *Service de Notifications* du B-WARN.

En effet, *SQL Server 2005 Notifications Services* est une plate-forme de développement et de déploiement d'applications qui génèrent et envoient des notifications et des messages personnalisés à plusieurs abonnés. Elle sert à créer, d'une façon simple, des instances de notifications dans la base de données, avec plusieurs tables et déclencheurs (Triggers) nécessaires au fonctionnement. Une instance de *Notification Services* collecte des événements et des abonnements inscrits et génère des notifications puis les distribue à des services de remise externe tels qu'un serveur SMTP.

Avant le début du développement du module *Service de Notification* de B-WARN, PIXEL a décidé de surclasser ses bases de données du SQL Server 2005 à SQL Server 2008. Cette décision a été de même appliquée sur la base de données de B-WARN. Ce surclassement a été réalisé sans faire attention que la plate-forme *Notifications Services* est retirée du SQL Server 2008.

Après avoir remarqué l'absence du *Notification Services* et afin de pouvoir intégrer dans B-WARN les fonctionnalités de notification, l'équipe de développement fut obligé de trouver

une solution alternative à l'utilisation de cette plate-forme. La solution a été de créer manuellement des déclencheurs qui génèrent les messages de notifications après chaque événement.

La solution trouvée a été réalisable et a bien rendu les résultats souhaités. Mais le temps de développement et des tests unitaires nécessaires à cette solution a été à peu près le double du temps prévu dans la planification initiale du projet. Ceci a implicitement retardé l'intégration du module *Services de Notifications*.

1.2.2 Aspect graphique de B-WARN

L'aspect graphique de B-WARN qui comporte la couleur de fond, la police des caractères et tous les autres paramètres visuels de l'application a été un sujet argumentatif entre les personnes qui participent à la conception de l'application. Certains ont suggéré d'utiliser l'aspect de *Microsoft Office 2007* avec les couleurs gradient Blue, Noir et Argent, et l'aspect de *Ribbon* pour l'affichage du menu. D'autres ont recommandé l'utilisation de l'aspect de *Microsoft Windows Vista* avec ses effets de transparence. Et aussi, la création d'un aspect graphique propre à l'application a été de même recommandée par un membre de l'équipe.

Tout au long du développement du projet, plusieurs essais ont été réalisés sans aboutir à un aspect accepté par tous les membres.

Enfin, puisque B-WARN serait utilisé sur les différentes versions du système d'exploitation Windows (XP, Vista, Windows 7), les interfaces graphiques ont été développées de sorte d'utiliser les paramètres graphiques, selon la version Windows installée sur la station de l'utilisateur, et toutes les modifications graphiques personnalisées par ce dernier. De cette façon, il était sûr que l'aspect de l'application sera confortable pour les utilisateurs. En effet, ces derniers ont eux mêmes choisi l'aspect de leurs systèmes d'exploitation.

L'aspect de l'application n'a pas été finalisé jusqu'aux dernières semaines de développement de l'application, ce qui n'était pas attendu et qui a fait perdre pas mal temps de développement.

1.2.3 Cycle de vie des tickets

La conception détaillée de B-WARN a comporté la création d'un cycle de vie des tickets flexible et personnalisable. Ce cycle de vie consiste à créer un nombre non limité des états des tickets, et par la suite spécifier les conditions de transition d'un état à un autre, ainsi que toutes les fonctions de traitement des tickets autorisés pour chaque état. Cette fonctionnalité a été décidée pour ne pas limiter le cycle de vie des tickets, et pour laisser le choix de ce cycle à l'entreprise qui utilise le système selon ses besoins.

Pendant la phase de développement de B-WARN, surtout au début du codage de module *Gestionnaire des tickets*, la réalisation du cycle de vie décidé semblait être trop compliqué et nécessitait un grand temps de développement et de tests, relativement non mérité par cette fonctionnalité, du fait que la première version de cette application serait utilisée seulement par PIXEL, ce qui rend inutile le cycle de vie flexible.

Cette déduction a nécessité une ré-conception de cette partie du système afin de transformer le cycle de vie des tickets de B-WARN en un cycle statique, prédéfini selon le workflow du service client à PIXEL. Cela a retardé la phase de codage du module en question.

1.2.4 Taille maximale des fichiers joints des emails

Après la fin du développement du *Windows Service*, responsable de la récupération automatique des emails et durant les phases des tests unitaires, les développeurs ont rencontrés une situation dans laquelle l'activité de ce service a été bloquée. Après plusieurs tests et débogage des procédures exécutés pour ce service, les développeurs ont trouvé des emails avec des pièces jointes de grande taille qui étaient en traine de ralentir et parfois bloquer la récupération des messages pendant leurs téléchargements.

A cause de ce problème, les concepteurs ont remarqué que la conception du module *Gestionnaire de Emails* n'a pas pris en considération les limites de téléchargement des fichiers du et vers les serveurs SMTP et POP3. C'est pourquoi, ils ont ajouté des paramètres qui spécifient la taille maximale de fichiers joints acceptés par le système. Ces paramètres sont initialisés par défaut à 3 Mo et ils pourront être modifiés par l'administrateur du système selon les paramètres de connexion internet de l'entreprise. Pour tout fichier dépassant cette

taille, le système notifiera les utilisateurs que les fichiers joints n'étaient pas téléchargés, et ils pourront les télécharger par leurs interventions manuelles.

Cette fuite de conception a nécessité des changements dans le système depuis la structure des tables de données, jusqu'aux interfaces graphiques en passant, par toutes les fonctions de manipulation de ces tables. Ce qui a exigé un temps de développement supplémentaire non planifié.

1.3 Problèmes techniques

Comme tout logiciel, les problèmes techniques ou complications de développement ont bien apparus au cours du développement du B-WARN SUITE. Ces problèmes n'avaient qu'une petite influence sur le déroulement du travail, et n'ont pas été nombreux. Dans ce qui suit, une description de deux problèmes techniques remarquables rencontrés durant le développement.

1.3.1 Choix entre RTF et HTML

Au cours de la conception du module *Gestionnaire des tickets* de B-WARN, les concepteurs ont choisi de sauvegarder les données représentant le corps ou la description du ticket en format RTF (Rich Text Format) afin de pouvoir changer la police des caractères (Font), et utiliser les caractères gras, italiques et aussi changer la taille et la couleur des caractères, cela pour permettre de mieux décrire les problèmes et les requêtes représentés par les tickets. Par la suite, au cours du développement, les programmeurs ont utilisé des champs de saisie texte spéciaux qui intègrent les fonctions de manipulation des données RTF.

Lors du début de développement de LIVE B-WARN, l'application web du système qui traite les mêmes données que le module de *Gestionnaire des tickets*, les développeurs ont rencontré un problème inattendu. C'était la difficulté de manipuler et visualiser les données en format RTF dans les pages de l'application web.

De même, les développeurs ont rencontré un problème en développant les fonctions et les procédures qui transforment les emails en tickets. En effet, le contenu des courriers électroniques pourrait être récupéré soit en format de texte intégral (Plain Text) soit en format HTML (Hyper Text Markup Language). Le premier format n'était pas un choix parce qu'il

prive le contenu du message de son police de caractères. Tandis que l'utilisation du deuxième remet en lumière le problème de conversion des textes du format HTML en RTF.

Après plusieurs essais de développement des procédures de conversion du format HTML en format RTF et vice versa, qui n'ont pas abouti à une solution acceptable, les développeurs ont décidé de changer le format de ces données descriptives des tickets du RTF au HTML. Facilement manipulé dans les pages web, ce format fournit les mêmes résultats que les formats RTF, et il est simplement traité par les applications Win. Ainsi les courriers électroniques seront simplement transformés en des tickets en préservant leurs polices de caractères.

Ce problème a été résolu par la modification du format des données du corps des tickets du RTF au HTML. Mais cela a nécessité un changement des interfaces graphiques déjà développées du module *Gestionnaire des tickets* qui traite ces données, et par conséquent une perte de temps non considérée.

1.3.2 Erreur SMTP code 5.7.1

Après le développement du module *Gestionnaire des Emails*, et au cours de la phase de test des fonctions d'envoi des courriers électroniques, une erreur est apparue, en remettant des messages à certains destinataires. C'était l'erreur SMTP code 5.7.1. « Relay is not Allowed ».

Cette erreur s'est manifestée en envoyant des courriers vers des comptes spécifiques comme le Hotmail et Gmail. C'était une erreur renvoyée par le serveur des destinataires déclarant que les courriers électroniques n'étaient pas acceptés par le serveur, mais rejeté. Après plusieurs recherches dans les différents forums et site informatiques, la meilleure définition de cette erreur était dans la page de dépannage de *Hotmail - Site Web Postmaster*. Sa définition est la suivante :

« **Code 550 DY-001**: Courrier refusé par Windows Live Hotmail par rapport à la politique Microsoft. Nous refusons généralement tout courrier en provenance d'une adresse IP dynamique, car ce type d'adresse n'est en principe pas utilisé pour la remise de courrier SMTP non authentifié à un serveur de messagerie Internet... » (24)

En d'autres termes, les serveurs de certains destinataires ont refusé de recevoir les courriers électroniques envoyés parce que l'IP de l'expéditeur, non publique, est anonyme. Pour des politiques et de raisons de sécurité ce type de courrier n'est pas acceptable.

Ce problème fut résolu par un simple changement de paramètre. En effet, afin de s'authentifier sur tous les serveurs sans aucun problème de sécurité, il était suffisant d'utiliser le nom du serveur SMTP du fournisseur du service internet ou l'ISP (Internet Service Provider). Par exemple, si le fournisseur est IDM, il faut utiliser *smtp.idm.net.lb*, tandis que s'il était SODETEL, il faut utiliser *smtp.sodetel.net.lb*.

Afin de trouver cette simple solution, j'ai consommé plusieurs heures de travail. Cela a été dû au changement de l'environnement du travail en effectuant les tests. En effet, le fournisseur du service internet au lieu de travail est différent de celui au domicile et aussi de celui à l'étranger. Pour cela, ce changement d'environnement a empêché de trouver rapidement la solution de l'erreur SMTP 5.7.1.

Malgré le temps imparti pour résoudre ce problème, ce dernier n'a pas influencé le déroulement du travail, vu qu'il a été limité dans une seule fonction du module. Ainsi, les processus de tests des autres fonctions ont été accomplis en même temps que la recherche de la solution.

2. Plan appliqué

Tous les problèmes rencontrés ont influencé d'une façon ou d'autre la planification chronologique du développement des tâches. En effet, ces problèmes ont affecté soit les dates du début du développement de certaines tâches, soit la période de développement désignée à quelques unes.

En général, les tâches les plus affectées ont été celles planifiées pour être traitées et développées en parallèle. Ceci était dû aux problèmes organisationnels rencontrés : la diminution des membres de l'équipe de développement et le changement des priorités de PIXEL. Seules les phases de tests des fonctions ont été réalisées en parallèle avec d'autres tâches parce qu'elles étaient réalisées par des développeurs du personnel de PIXEL qui ne faisaient pas partie de l'équipe de développement du B-WARN SUITE.

Le diagramme de GANTT suivant représente le plan appliqué tout au long du développement du projet. Il met en relief les différents retards et délais de chaque tâche et phase.

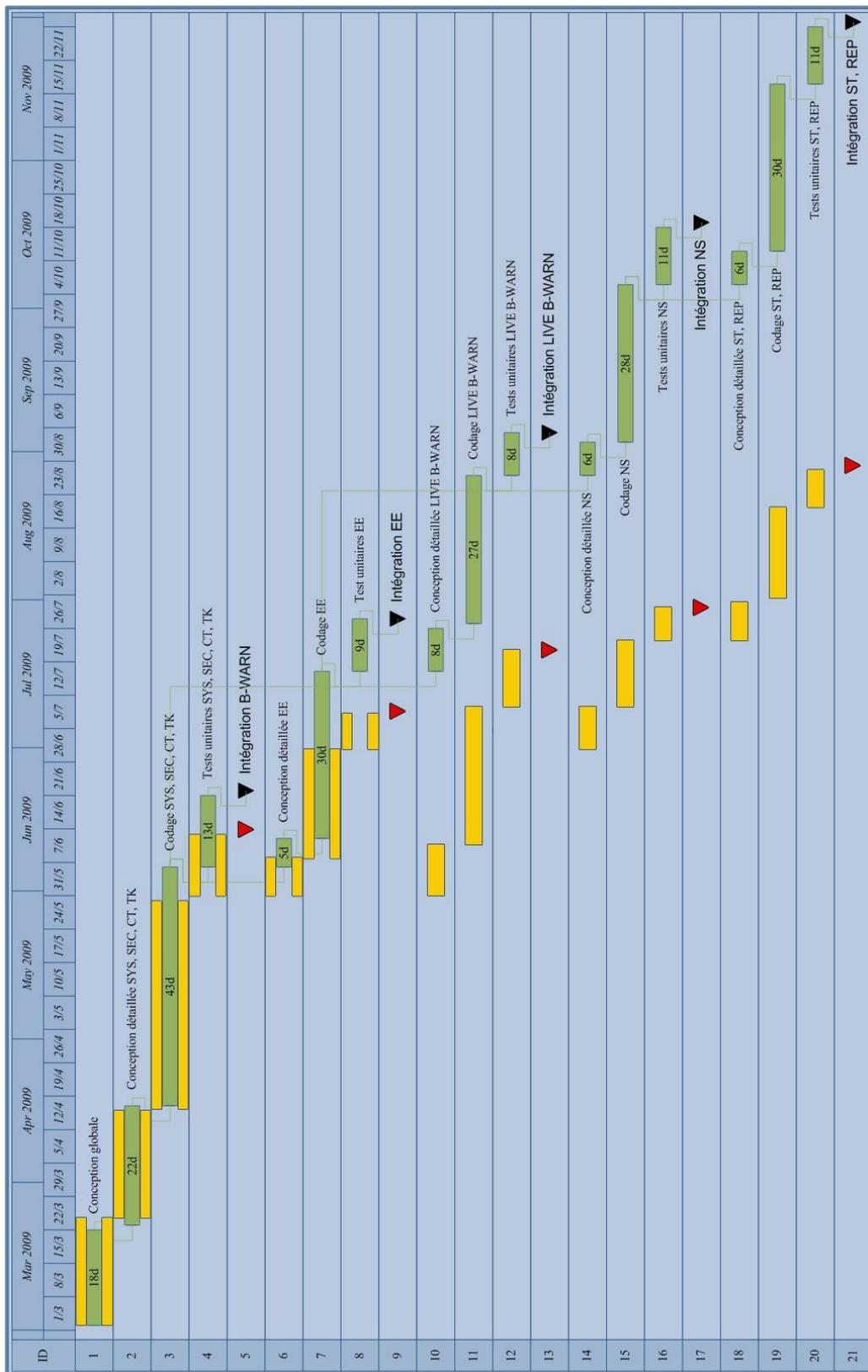


Figure 27 : Diagramme de GANTT représentant le plan de développement appliqué.

Le diagramme de GANTT de la Figure 27 montre clairement que les latences entre la période de développement planifié (coloré en jaune) et la période de développement effectif (coloré en vert) ont augmenté dès le mois de juillet pour atteindre 79 jours de travail, à peu près 3 mois en plus pour la dernière date d'intégration désignée. Cela a été dû principalement au changement de la priorité du projet qui a engendré la diminution successive des membres de l'équipe de développement.

De même, le temps de développement effectif du module *Service de Notifications* représenté dans le diagramme de GANT plus que le double de la période planifiée pour la raison de surclassent au SQL Server 2008 déjà expliqué.

3. Evaluation

Etant responsable du développement de B-WARN SUITE, j'en ai participé à toutes les phases, depuis le début de la conception détaillée du projet jusqu'à la dernière date d'intégration, en passant par toutes les phases de codage et de test de chaque fonctions du projet. De même, j'ai participé à la solution de la plupart des problèmes rencontrés, et parfois à leurs créations.

Interne pour PIXEL, ce projet n'a pas joui de la même importance que les autres projets de l'entreprise, ce qui a été clair quand l'administration de PIXEL a pris la décision de diminuer la priorité du projet pour arriver à prendre la décision de l'ajourner.

Par contre, à l'administration de l'entreprise, ce projet avait pour moi une grande priorité, non pas que j'aie été le chef du projet, mais parce que j'étais sûr et certain de son importance. Et que j'avais confiance en ses avantages et les bienfaits qu'il apporterait à l'entreprise afin de pouvoir résoudre le désordre du service client de l'entreprise, devenu insupportable. Pour cela, j'ai décidé de poursuivre seul le développement du projet quand l'entreprise a pris la décision d'ajourner le projet, tout en réalisant les surcharges que j'avais déjà et les efforts que je devrais fournir afin de l'achever.

En tant que projet d'équipe, je ne peux pas considérer B-WARN SUITE comme tel, sauf durant les trois premiers mois de son développement. Pendant cette période, les quatre personnes qui formaient l'équipe de développement ont été fort enthousiasmées, bien qu'ils fussent chargés d'autres responsabilités. En effet, B-WARN SUITE a été un nouveau sujet

pour l'équipe, différent des sujets traités d'habitude par PIXEL : gestion d'assurance, gestion financière, gestion de stock etc.

Tout au long de cette période, l'équipe de développement a manifesté un niveau professionnel, surtout durant les recherches réalisées, généralement laisser tombées dans la plupart des projets de PIXEL, vu que le personnel s'est familiarisé avec ces derniers et avec leurs objectifs et leurs fonctions traitées. Cette période a été pour moi une expérience de management et de conduite d'une équipe de développement, dès le premier jour du commencement du projet, ce qui n'était pas valable dans mes autres expériences de management à PIXEL qui ont été limités à un module du projet ou à la phase d'implémentation.

Dans la deuxième partie du développement du projet, qui a demandé une période approximative de six mois, j'ai effectué tout seul le travail restant qui a comporté des phases de conception et des phases de codage, à l'exception des phases de tests qui ont été réalisées par différents membres du personnel.

Cette partie ne représente pas une expérience de travail d'équipe. Par contre, elle a été pour moi un challenge pour achever le plus tôt possible et prouver l'importance de ce projet. En plus, cette partie du projet a formé une excellente expérience technique surtout durant le développement des fonctionnalités neuves comme la connexion au serveur SMTP et POP3 et leurs fonctions de récupération et d'envoi des courriers électroniques.

Du point de vue technique, cette partie m'a fourni une expérience technique remarquable et j'ai pu assurer l'achèvement du projet. Mais de point de vue professionnel, je crois qu'il était préférable de poursuivre le développement du projet par un travail d'équipe et non pas par un travail individuel, ce qui aurait pu diminuer la période de développement et augmenter peut être les innovations et la créativité des fonctionnalités de B-WARN SUITE.

Enfin, d'après mon évaluation personnelle et mes observations du déroulement du travail, de toutes les décisions prises et les problèmes rencontrés, j'ai pu révéler les points faibles et les points forts qui ont affectés le développement. Ces différents points sont résumés dans le tableau suivant (Table X) :

Table X : Points faibles et points forts du développement de B-WARN SUITE.

Description	Points faibles	Points forts
Achèvement du travail individuellement		X
Choix des technologies	X	
Conduite de projet et répartition des tâches (pendant les trois premiers mois)		X
Conduite de projet et répartition des tâches (pendant les six premiers mois)	X	
Disponibilité des ressources matérielles		X
Enthousiasme de l'équipe (durant les trois premiers mois)		X
Expériences technique		X
Sous traitement du projet par rapport aux autres projets de l'entreprise	X	
Surcharges externes des membres de l'équipe	X	
Temps de développement limité	X	

Enfin, la situation dans laquelle B-WARN SUITE a été développée a sérieusement affecté le déroulement du travail. En plus, les problèmes rencontrés ont été incontrôlés et inattendus par leurs majorités. Pour cela, les résultats de ce projet, par rapport à tous ces derniers, ont été de ma part, acceptables, mais ils pourraient être meilleurs dans d'autres circonstances.

Conclusion

B-WARN SUITE, d'après les fonctionnalités qu'il incorpore, est catégorisé comme système de suivi des problèmes (ou bugs), et un système limité de gestion du service client. Il comporte deux applications : B-WARN et LIVE B-WARN.

La première, application bureautique, utilisée par les employés. Elle intègre les fonctions de manipulation des tickets d'incidents et leurs escalades d'un niveau de service à un autre. Comme elle incorpore la fonction de récupération automatique des courriers électroniques et leur transformation en tickets de suivi.

De même, elle offre des fonctions de sécurité avancées qui aident à la protection des données, et des fonctions de notifications qui servent à avertir les employés le plus tôt possible de leurs tâches de service.

En outre, elle comporte des outils et des rapports statistiques qui permettent à l'administration d'exploiter les données et de les transformer en des sorties utiles qui aident à la supervision des différentes activités du service client.

La deuxième, application web, utilisée par les clients. Elle comporte des fonctions limitées de manipulation des tickets de suivi, et de supervision de leurs progressions tout au long de leurs traitements. En plus, elle contient des rapports et des outils statistiques qui aident les clients à afficher les états de leurs tickets.

Actuellement, B-WARN SUITE est mis en service et effectue toutes les fonctions recommandées et souhaitées par l'administration de PIXEL. Malgré ses fonctionnalités, ce système reste un système de gestion du service client, qui nécessite plusieurs modifications pour englober les fonctions de gestion de la relation client. Cette dernière, est visée par toute entreprise ayant l'intention de conserver son portefeuille en offrant à ses clients des services à haut niveau, et attirer de nouveaux clients.

Pour cela, cette première version du système doit être suivie par une autre le plus tôt possible, pour regrouper toutes les fonctionnalités des systèmes de gestion de la relation client. Cette future version pourra comporter plusieurs fonctions susceptibles d'améliorer le système et d'augmenter ses bienfaits :

Conformité avec ITIL v2 *Support service*

- La distinction entre les incidents et les problèmes.
- La planification et l'organisation des différentes activités de développement et d'implémentation réalisées par le personnel de l'entreprise.
- La gestion des mises à jour des différentes versions des produits développés par l'entreprise.

Flexibilité

- Un cycle de vie flexible des tickets, pouvant être changé et personnalisé selon les besoins de l'entreprise.
- Une amélioration de la structure hiérarchique des utilisateurs du système, comportant la création de plusieurs entités hiérarchiques afin de pouvoir représenter tous les départements de l'entreprise concernés par la relation client.

Comptabilité et gestion financière

- Une gestion avancée des contrats de niveau du service (SLA), comportant l'administration financière des coûts de chaque contrat et de chaque requête additionnelle recommandée par les clients. Ce qui permet la liaison du coût de service au temps de service effectif.
- Une gestion comptable complète gérant tous les comptes de l'entreprise, les paiements, les dettes des clients, les relevés des comptes, et toute autre fonction de gestion financière nécessaire.

Service client avancé

- Une base de données publique accessible par les clients pour résoudre les problèmes fréquemment rencontrés, et pour répondre aux questions fréquemment posées.
- La discussion en ligne servant à communiquer instantanément avec les clients.

Optimisation de la performance

- Une fonction d'archivage qui sera utilisée chaque an ou plus, pour archiver les anciennes données, et par la suite optimiser la recherche et l'extraction des données.

Plusieurs autres fonctionnalités pourront être intégrées, mais cela dépend des ressources disponibles, et certainement du temps consacré au développement de cette nouvelle version.

B-WARN SUITE a été le premier système dont j'ai été responsable hors des sujets traditionnels développés par PIXEL. En tant que chef de projet, c'était une opportunité pour pratiquer mes connaissances acquises durant mes cours universitaires, et surtout ceux de la conduite des projets et du management humain.

L'application de ces connaissances était plus difficile que j'ai imaginé. En effet, les contraintes imposées par la situation au cours de laquelle le projet a été développé, n'ont pas permis l'application de toutes les étapes du cycle de développement du logiciel. Par exemple, le choix des technologies a été dépassé du fait que ce projet n'a pas un rendement financier direct, et l'entreprise a décidé de minimiser son coût, ce qui a impliqué l'utilisation des ressources matérielles et logiciels existantes.

Enfin, c'était pour moi une expérience exceptionnelle, qui m'a offert l'occasion de développer mes compétences techniques, en créant plusieurs fonctions et procédures non utilisées d'habitude dans les systèmes développés par PIXEL. En plus, c'était une opportunité pour expérimenter mes capacités de traiter et de gérer un sujet inhabituel, avec lequel je ne m'étais pas familiarisé, et de traiter tous ses points essentiels selon les exigences de l'entreprise. Sans oublier l'expérience de conduite du projet qui était ma première de son type.

Finalement, le plus important de ce projet c'était les connaissances que j'ai acquises à propos du service client et de la relation client, leurs avantages et leurs portées sur les entreprises. La perception de ces titres, qui ont été pour moi des sujets secondaires, a eu un impact fort positif sur ma progression professionnelle et ma perspective de mon parcours d'ingénieur.

Annexes

Annexe 1 : Le premier bug informatique

« Bug » un mot d'origine anglais qui signifie « Insecte » est francisé en « Bogue ». Ce dernier utilisée au Québec et recommandé en France par la DFLGLG (Délégation générale à la langue française et aux langues de France) en genre masculin. En informatique, « Bug » est utilisée pour désigner un défaut de logique d'un programme informatique qui provoque un dysfonctionnement ou un traitement anormal et inattendu.

La première utilisation du mot pour décrire un défaut de système (non informatique) revient à Thomas EDISON qui l'a utilisé dans une lettre envoyée à son collègue en 1878 :

« It has been just so in all of my inventions. The first step is an intuition, and comes with a burst, then difficulties arise — this thing gives out and [it is] then that 'Bugs' — as such little faults and difficulties are called — show themselves and months of intense watching, study and labor are requisite before commercial success or failure is certainly reached. » (25).

En 1947, vice-amiral Grace Murray HOPPER, informaticienne américaine et officier marine des Etats –Unis, après sa libération du service militaire, était en train de travailler sur le calculateur « Mark II Aiken Relay » de l'université de Harvard.

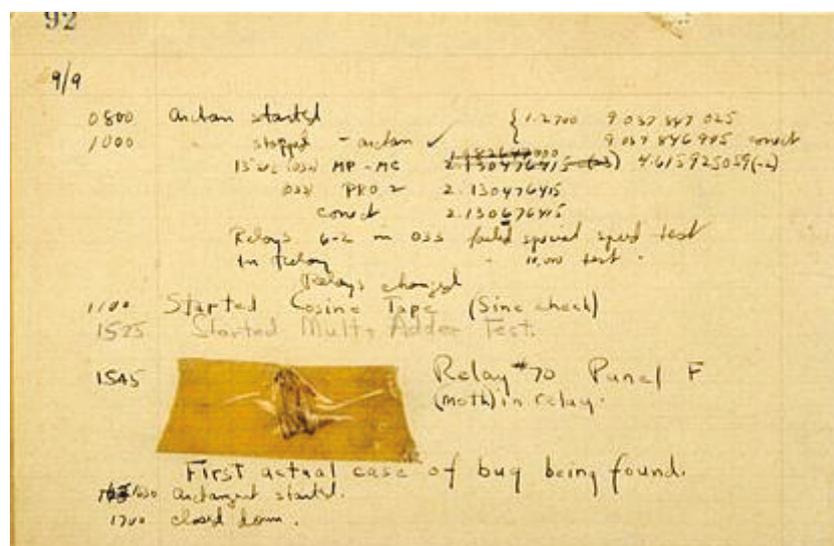


Figure 28 : Journal de bord, contenant ce qui est peut-être le premier bug informatique trouvé.

Le 9 Septembre 1947, après l'investigation d'un dysfonctionnement de la machine, un des techniciens de son équipe a trouvé un insecte (Bug) piégé dans le relais #70 du tableau F.

L'insecte a été enlevé, affiché dans le journal de bord et libellé « First actual case of computer bug » ou le premier cas réel d'un bug d'ordinateur (26).

Le journal de bord affiché dans la Figure 28 est complètement exposé dans l'*Institute Smithsonian* avec le Bug attaché (27).

L'utilisation du mot « Actual » (actuel) par le technicien montre que ce dernier s'était familiarisé avec le terme Bug, ce qui crée des doutes sur le fait que c'était le *Premier Bug Informatique*. Pourtant, ce cas reste le plus ancien d'un bug informatique jamais enregistré.

Annexe 2 : Des bugs informatiques catastrophiques

Les bugs informatiques ne sont pas tous provoqués par des insectes, mais aussi dus à des fautes humaines, nées des mains d'hommes. Ils peuvent être parfois négligeables, comme ils peuvent coûter cher.

Quelques exemples des bugs historiques et catastrophiques (28):

- La fusée Ariane 5 a explosée 40 secondes après le décollage. Cet incident catastrophique qui a eu lieu le 4 juin 1996 est dû à un bug informatique. En effet, dans la nouvelle fusée les ingénieurs avaient introduit le logiciel d'Ariane 4 qui était bien différent. Ce logiciel n'était capable de calculer les données d'Ariane 5.
- Entre 1985 et 1987 le Therac-25, appareil de radiothérapie, a coûté la vie à plusieurs patients dans divers hôpitaux des Etats-Unis à cause d'une erreur informatique. Ce bug a permis à l'appareil d'envoyer aux patients des rayons X 100 fois supérieurs au prévu.
- En 1980, le Commandement de la défense aérospatiale de l'Amérique du Nord (NORAD) a indiqué que les Etats-Unis sont attaqués par de missiles. Le problème était dû à une probabilité que le logiciel des rapports n'a pas prise en considération. En 1983, un satellite soviétique rapporte un lancement des missiles US, mais le responsable a décidé de considéré ce fait comme une fausse alarme, ce qui aurait déclenché une troisième guerre mondiale.

Dans ces domaines critiques, les conséquences d'un bug informatique pourraient être monstrueuses. Pour cela, les responsables essayent de contrôler la qualité à travers des certifications et des normes internationales qui décrivent le processus de développement et de test d'un logiciel.

Malheureusement, dans des domaines moins critiques, le principe de « Logiciel banane » prédomine encore. Les entreprises vendent leurs logiciels immatures malgré les tests insuffisants, et pensent qu'ils feront un meilleur profit. Agir de cette façon peut leur coûter cher et les faire même disparaître du marché.

Pour éviter les conséquences des bugs informatiques, les entreprises font traverser leurs logiciels par des processus de test intensifs. Jusqu'à nos jours, un logiciel qui contient 2 bugs dans 1000 lignes de codes est considéré comme un bon logiciel. Pour Windows XP par exemple qui contient 40 millions de lignes de codes, selon les statistiques, environ 800,000 bugs sont glissés dans le logiciel.

Annexe 3 : La fonction stockée fnGetTicketAssignedTo(int,date)

La fonction `fnGetTicketAssignedTo(int,date)` est responsable de chercher dans l'historique du ticket en paramètre, et de récupérer le nom de l'employé auquel un ticket spécifique est assigné dans une date donnée.

```
CREATE FUNCTION [fnGetTicketAssignedTo2] (@TicketID int, @Date as date)
RETURNS nvarchar(100)
AS
BEGIN

DECLARE @AssignedTo nvarchar(100)

SET @AssignedTo = (
    SELECT TOP(1) SEC_SystemUsers.UserName
    FROM TK_TicketAssigns
    LEFT JOIN SEC_SystemUsers ON TK_TicketAssigns.AssignedTo
                                = SEC_SystemUsers.SystemUserID
    WHERE TK_TicketAssigns.TicketID = @TicketID
    AND TK_TicketAssigns.AssignedOn <= @Date
    ORDER BY TK_TicketAssigns.AssignedOn DESC, TK_TicketAssigns.SysID DESC
)

RETURN @AssignedTo

END
```

Annexe 4 : Le déclencheur Trig_TK_Tickets_Insert

Le déclencheur Trig_TK_Tickets_Insert est déclenché à l'insertion d'une nouvelle ligne de donnée dans la table TK_Tickets. Il est responsable de générer un numéro de série du ticket, et d'insérer l'état initial *NEW* du ticket dans la table TK_TicketStatuses.

```
CREATE TRIGGER [dbo].[Trig_TK_Tickets_Insert] ON [dbo].[TK_Tickets]
FOR INSERT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DECLARE @ReportedOn DateTime
        SELECT @ReportedOn = ReportedOn FROM inserted
        DECLARE @SerialKey nvarchar(250)
        DECLARE @Serial nvarchar(250)
        DECLARE @NewSerialNo nvarchar(250)
        -- Retrieve last serial no
        SET @SerialKey = 'TK-' + CAST(YEAR(@ReportedOn) AS nvarchar)
        EXEC [SYS_GetSerialNo] @SerialKey, @Serial OUTPUT
        SET @Serial = REPLACE(STR(@Serial, 6), ' ', '0')
        SET @NewSerialNo = @Serial + '-' +
            SUBSTRING(CAST(YEAR(@ReportedOn) AS nvarchar), 3, 2)
        -- Update Ticket # in inserted row
        UPDATE TK_Tickets SET TicketNo = @NewSerialNo
        FROM TK_Tickets
        INNER JOIN inserted ON TK_Tickets.TicketID = inserted.TicketID
        -- Insert Default ticket status [21 Status for NEW]
        INSERT INTO [TK_TicketStatuses] ([TicketID], [StatusID],
            [TimeCostInMin], [Notes], [CreatedBy])
        SELECT TicketID, 21, 0, NULL, CreatedBy
        FROM inserted
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
        BEGIN
            ROLLBACK TRANSACTION;
        END
    END CATCH;
END
```

Annexe 5 : Classe Projects du projet SYS.DataAccess

La classe `Projects` du projet `SYS.DataAccess` est responsable de récupérer les données de la table de données `SYS_Projects`, et de sauvegarder les données insérées par l'utilisateur dans la même table. Ces activités sont réalisées respectivement à l'aide de deux fonctions : `GetData()` et `Update()`. La première, renvoi un objet `DsProjects` de type `DataSet` qui contient les données récupérées. La deuxième, prend en paramètre un objet `DsProjects` qui contient les données insérées ou modifiées par l'utilisateur et les sauvegarde dans la base de données. Les fonctions de cette classe sont appelées par les fonctions du projet `BusinessRules` du même espace de nom.

```
Public Class Projects

    Private m_ConnectionString As String
    Private m_ErrorOccured As Boolean = False
    Private m_ErrorMessage As String
    Private m_Trans As SqlClient.SqlTransaction

    Public WriteOnly Property ConnectionString() As String
        Set(ByVal Value As String)
            m_ConnectionString = Value
        End Set
    End Property

    Public ReadOnly Property ErrorOccured() As Boolean
        Get
            Return m_ErrorOccured
        End Get
    End Property

    Public ReadOnly Property ErrorMessage() As String
        Get
            Return m_ErrorMessage
        End Get
    End Property

    Public Function GetData() As SYS.DataSets.DsProjects
        Dim ds As New SYS.DataSets.DsProjects

        m_ErrorOccured = False
        m_ErrorMessage = ""

        Me.SqlConnection1.ConnectionString = m_ConnectionString

        Try
            Me.SqlDataAdapter1.Fill(ds, ds.SYS_Projects.ToString)

        Catch e As Exception
            m_ErrorOccured = True
            m_ErrorMessage = e.Message
        End Try
    End Function
End Class
```

```

        Return ds
    End Function

Public Function GetData(ByVal _szSql As String) As SYS.DataSets.DsProjects
    Dim ds As New SYS.DataSets.DsProjects

    m_ErrorOccured = False
    m_ErrorMessage = ""

    Me.SqlConnection1.ConnectionString = m_ConnectionString
    Me.SqlDataAdapter1.SelectCommand.CommandText = _szSql

    Try
        Me.SqlDataAdapter1.Fill(ds, ds.SYS_Projects.ToString)
    Catch e As Exception
        m_ErrorOccured = True
        m_ErrorMessage = e.Message
    End Try

    Return ds
End Function

Public Sub Update(ByVal _ds As SYS.DataSets.DsProjects)

    Me.m_ErrorOccured = False
    Me.m_ErrorMessage = ""

    Me.SqlConnection1.ConnectionString = Me.m_ConnectionString
    Me.SqlConnection1.Open()

    Me.m_Trans = Me.SqlConnection1.BeginTransaction()

    Me.SqlInsertCommand1.Transaction = Me.m_Trans
    Me.SqlDeleteCommand1.Transaction = Me.m_Trans
    Me.SqlUpdateCommand1.Transaction = Me.m_Trans

    Try

        Me.SqlDataAdapter1.Update(_ds, _ds.SYS_Projects.ToString)
        _ds.AcceptChanges()
        Me.m_Trans.Commit()

    Catch e As System.Exception
        Me.m_ErrorOccured = True
        Me.m_ErrorMessage = e.Message

        Me.m_Trans.Rollback()
        _ds.Clear()
    Finally
        Me.SqlConnection1.Close()
    End Try
End Sub

End Class

```

Annexe 6 : Classe ServiceLevels du projet SYS.BusinessRules

La classe `ServiceLevels` du projet `SYS.BusinessRules` est responsable de manipuler les données de la table des niveaux de service. Elle comporte deux fonctions essentielles pour ces activités : `GetData()` et `Update()`. Ces deux fonctions appellent les mêmes fonctions existant dans le projet `DataAccess` du même espace de nom. En plus, cette classe contient une fonction `SetDefaultValues(...)` responsable d'initialiser l'objet `DataSet` en paramètre, et une fonction `IsDataValid(...)` responsable de valider les données avant de mettre à jour la base de données. Les fonctions de cette classe sont appelées par les classes des projets UI du même espace de nom.

```
Public Class ServiceLevels

    Private m_ErrorOccured As Boolean = False
    Private m_ErrorMessage As String

    Public ReadOnly Property ErrorOccured() As Boolean
        Get
            Return m_ErrorOccured
        End Get
    End Property

    Public ReadOnly Property ErrorMessage() As String
        Get
            Return m_ErrorMessage
        End Get
    End Property

    Public Function GetData() As SYS.DataSets.DsServiceLevels
        Dim oCDA As New SYS.DataAccess.ServiceLevels
        oCDA.ConnectionString = [Shared].Variables.ConnectionString

        GetData = oCDA.GetData()
        Me.m_ErrorMessage = oCDA.ErrorMessage
        Me.m_ErrorOccured = oCDA.ErrorOccured
    End Function

    Public Sub Update(ByVal _ds As SYS.DataSets.DsServiceLevels)
        Dim oCDA As New SYS.DataAccess.ServiceLevels
        oCDA.ConnectionString = [Shared].Variables.ConnectionString

        If DataIsValid(_ds) Then
            oCDA.Update(_ds)

            Me.m_ErrorOccured = oCDA.ErrorOccured
            Me.m_ErrorMessage = oCDA.ErrorMessage
        End If
    End Sub
```

```

Private Function DataIsValid(ByVal _ds As SYS.DataSets.DsServiceLevels) As Boolean
    Dim szValidationErrors As String = ""
    Me.m_ErrorOccured = False
    Me.m_ErrorMessage = ""

    For i As Integer = 0 To _ds.SYS_ServiceLevels.Rows.Count - 1
        With _ds.SYS_ServiceLevels.Rows(i)
            .ClearErrors()
            If .RowState <> DataRowState.Deleted And .RowState <>
DataRowState.Unchanged Then
                If .Item("ServiceLevelRank") <= 0 Then
                    .SetColumnError("ServiceLevelRank", "Rank should be > 0.")
                    szValidationErrors &= "Rank should be > 0." & vbCrLf
                End If

                If IsDBNull(.Item("ServiceLevelCode")) OrElse .Item("ServiceLevelCode")
= "" Then
                    .SetColumnError("ServiceLevelCode", "Code is a required field.")
                    szValidationErrors &= "Code is a required field." & vbCrLf
                End If

                If IsDBNull(.Item("ServiceLevelDesc")) OrElse .Item("ServiceLevelDesc")
= "" Then
                    .SetColumnError("ServiceLevelDesc", "Description is a required
field.")
                    szValidationErrors &= "Description is a required field." & vbCrLf
                End If
            End If
        End With
    Next

    If szValidationErrors.Trim <> "" Then
        Me.m_ErrorOccured = True
        Me.m_ErrorMessage = "Errors found while validating data:" & vbCrLf &
szValidationErrors
        Return False
    Else
        Return True
    End If
End Function

Public Sub SetDefaultValues(ByVal _ds As SYS.DataSets.DsServiceLevels)
    With _ds.SYS_ServiceLevels
        .Columns("ServiceLevelID").AutoIncrementSeed = -1
        .Columns("ServiceLevelID").AutoIncrementStep = -1
        .Columns("ServiceLevelCode").AllowDBNull = True
        .Columns("ServiceLevelDesc").AllowDBNull = True
        .Columns("ServiceLevelRank").DefaultValue = 10
    End With
End Sub

End Class

```

Bibliographie

1. *Le grand dictionnaire terminologique*, [En ligne] Office québécois de la langue française, <http://www.granddictionnaire.com>.
2. BAGOZZI R., 1975. Marketing as Exchange, *Journal of Marketing*, **39**, 32 - 39.
3. PEELEN E., JALLAT F., STEVENS E., VOLLE P., 2006. *Gestion de la relation client*, Pearson Education, ISBN 9782744071515.
4. *Optimisation de la gestion des interactions client avec internet et le mobile*, [En ligne] MARKESS International, <http://www.markess.fr/demandedocument.php?refdoc=642>.
5. FISK P., 2008. *Marketing genius*, Capstone Publishing Limited, ISBN 1841126810.
6. BRUSSET O., 2001. La GRC, tout le monde en fait... Ou va en faire, *Marketing Direct*, **52**.
7. TURBAN E., 2006. *Electronic Commerce: A Managerial Perspective*, Prentice Hall, ISBN 0131854615.
8. *10 Minute Guide - Customer service programme*, [En ligne] The Chartered Institute of Marketing, <http://www.cim.co.uk/filestore/resources/10minguides/customerservice.pdf>.
9. *Customer Service, Product Selection Falls Short of Shopper's Expectations*, [En ligne] CRM Today, <http://www.crm2day.com/news/printnews.php?id=119766>.
10. PEPPERS D., ROGERS M., 2008. *Rules to Break and Laws to Follow*, Hoboken, N.J. : John Wiley & Sons, ISBN 9780470227541.
11. *The Global Standard for Customer Service*, [En ligne] The International Customer Service Institute, <http://www.ticsi.org/docs/brochure/ticss.pdf>.
12. *Delivering High Availability Services Using a Multi-Tiered Support Model*, [En ligne] Windley's Technometria, <http://www.windley.com/docs/Tiered%20Support.pdf>.
13. *Excellent Customer Service and the pursuit of profit - the role of better business operations*, [En ligne] Celerant Consulting, <http://www.celerantconsulting.com/Downloads/Perspectives/Celerant%20Perspective%20-%20Excellent%20Customer%20Service.pdf>.

-
14. *Mauvais service client, 338,5 milliards de dollars de perte par an*, [En ligne] docstoc, http://www.docstoc.com/docs/DownloadDoc.aspx?doc_id=19423898.
 15. *Better Consumer Relations With Trouble Ticket Software*, [En ligne] Content 4 Print, <http://www.content4reprint.com/business/customer-service/better-consumer-relations-with-trouble-ticket-software.htm>.
 16. *Bug classification based on severity and priority*, [En ligne] Software Testing Concepts, <http://www.softwaretestingconcepts.com/bug-classification-based-on-severity-and-priority>.
 17. *Cycle de vie d'un bug sous Bugzilla*, [En ligne] QA Testing, http://www.qatesting.be/index.php?option=com_content&task=view&id=45&Itemid=2.
 18. *Software Testing - Bug Life Cycles*, [En ligne] Buzzle.com, <http://www.buzzle.com/editorials/4-6-2005-68177.asp>.
 19. *Help Desk Pilot*, [En ligne] <http://www.helpdeskpilot.com>.
 20. *OMNINET*, [En ligne] <http://www.omnitacker.biz>.
 21. *NetResults Tracker*, [En ligne] <http://www.netresultstracker.com>.
 22. *Kayako helpdesk*, [En ligne] <http://www.kayako.com>.
 23. *Bugzilla*, [En ligne] <http://www.bugzilla.org>.
 24. *Dépannage*, [En ligne] Windows Live, <http://mail.live.com/mail/troubleshooting.aspx#Codes>.
 25. HUGES T., 1989. *American Genesis*, Penguin Books, ISBN 0140097414.
 26. *First Computer Bug*, [En ligne] James S. Huggins' Refrigerator Door, http://www.jamesshuggins.com/h/tek1/first_computer_bug.htm.
 27. *Log Book With Computer Bug*, [En ligne] National Museum of American History, <http://americanhistory.si.edu/collections/object.cfm?key=35&objkey=30>.
 28. *10 historical software bugs with extreme consequences*, [En ligne] Royal Pingdom, <http://royal.pingdom.com/2009/03/19/10-historical-software-bugs-with-extreme-consequences/>.

Liste des figures

Figure 1 : Structure organisationnelle utilisée au cours du développement.....	13
Figure 2 : Structure organisationnelle utilisée au cours du service après vente.....	14
Figure 3 : Résultats de l'étude « Optimisation de la gestion des interactions clients avec internet et le mobile » réalisée par MARKESS International.	27
Figure 4 : Le modèle « 5 P's of Customer Service » de TICSI.	30
Figure 5 : Modèle de service Multi-tiers à 5 niveaux.	31
Figure 6 : Résultats de l'étude « The Cost of poor customer service : the economic impact of the customer experience and engagement.....	33
Figure 7 : Cycle de vie d'un bug selon Bugzilla.	40
Figure 8 : Cycle de vie d'un bug selon Buzzle.com.....	41
Figure 9 : Différentes méthodes pour créer un ticket dans B-WARN SUITE.....	53
Figure 10 : Processus de traitement d'un ticket et son escalade entre les différents niveaux de services.	54
Figure 11 : Diagramme de GANTT du développement du projet.	59
Figure 12 : Fenêtre de saisie des priorités dans B-WARN.....	63
Figure 13 : Gestion de priorités dans B-WARN à deux critères.....	64
Figure 14 : Composition graphique de l'interface principale de B-WARN.....	66
Figure 15 : Diagramme de classes du module de sécurité.....	69
Figure 16 : Diagramme de cas d'utilisation du module "Registre des contacts".....	71
Figure 17 : Cycle de vie d'un ticket dans B-WARN.....	73
Figure 18 : Diagramme de classes du module "Gestionnaire des Tickets".....	74
Figure 19 : Diagramme de séquences : La récupération des messages du serveur POP3 dans B-WARN.....	78
Figure 20 : Diagramme d'activités du traitement d'un email récupéré par B-WARN.....	79
Figure 21 : Instance de notification conditionnelle de B-WARN.....	80
Figure 22 : Diagramme de séquences du processus de validation des instances de notification dans B-WARN.....	81
Figure 23 : Fenêtre de saisie <i>Charts Manager</i> de B-WARN.....	83
Figure 24 : Diagramme de cas d'utilisation d'LIVE B-WARN.....	86
Figure 25 : Diagramme de composants de la structure physiques de B-WARN.....	91
Figure 26 : Réseau et installations nécessaires pour le déploiement de B-WARN SUITE.....	94
Figure 27 : Diagramme de GANTT représentant le plan de développement appliqué.....	105
Figure 28 : Journal de bord, contenant ce qui est peut-être le premier bug informatique trouvé.....	113

Liste des tableaux

Table I : Les objectifs dégagés par l'administration de l'entreprise.	21
Table II : Comparaison entre marketing Transactionnel et Relationnel.	26
Table III : Classification des Bugs par sévérité.	38
Table IV : Association de priorité par sévérité et fréquence du bug.	39
Table V : Les catégories des systèmes disponibles trouvés.	44
Table VI : Les types des applications des systèmes disponibles trouvés.	45
Table VII : Les canaux de communication couverts par les systèmes disponibles trouvés.	46
Table VIII : Fonctionnalités intégrées dans les systèmes disponibles trouvés.	47
Table X : Configuration des machines requise pour le déploiement de B-WARN SUITE.	94
Table XI : Points faibles et points forts du développement de B-WARN SUITE.	108

RESUME

Dès l'apparition du concept marketing relationnel, la relation client-entreprise est devenue un facteur compétitif que les entreprises visent à gérer de la meilleure façon afin de conserver leurs clients et séduire des nouveaux. Cette gestion vise à intégrer un multi-canal qui couvre les divers canaux des communications client-entreprise, qui ont notamment augmenté avec l'évolution des technologies de télécommunications.

Le service client étant un point essentiel englobé par la gestion de la relation client, les systèmes de suivi des problèmes sont devenus fréquemment utilisés, notamment dans les entreprises qui offrent des services informatiques.

Dans leur majorité, ces systèmes utilisent le principe de ticket de suivi, afin de poursuivre la progression des bugs (ou problèmes) tout au long de leur traitement. En outre, ils intègrent une architecture de service à multi-tiers, utilisée par les entreprises pour offrir des services de haut niveau, et diminuer la perte du temps du personnel.

PIXEL, comme toute entreprise, a visé à développer un tel système pour organiser son département de service client et offrir à ses clients une prestation de haut niveau. Le développement de ce système dont j'ai été responsable, a été conçu pour englober tous les points susmentionnés.

Mots clés : GRC, Multi-canal, Service client, Bug, Ticket, Service multi-tiers.

SUMMARY

Since the relational marketing concept appeared, customer-enterprise relation became a competitive agent that all companies aim to manage and improve in order to maintain their customers and attract new ones. This management integrates a multi-channel which covers several customer-enterprise communication channels that have obviously increased with the telecommunication technologies evolution.

Customer service is a major topic covered by the customer relationship management. For this reason, issues tracking systems became widely used, especially in companies offering IT services.

The majority of these systems use the trouble ticket principle in order to follow the bug (or problem) progression during its processing cycle. Further, these systems integrate a multi-tiers service architecture, which is commonly used by enterprises in order to offer high availability services, and decrease their team time wasting.

Like all companies, PIXEL required to develop a similar system to organize its customer service department and offer its clients a high availability services. This system, which I was in charge of all its development process, was planned to cover all above mentioned topics.

Keywords: CRM, Multi-Channel, Customer service, Bug, Ticket, Multi-tiers support.