

Conception et développement de Gecko : logiciel de gestion de l'activité et de la connaissance d'une équipe de support informatique

Elina Lawson-Body

► To cite this version:

Elina Lawson-Body. Conception et développement de Gecko : logiciel de gestion de l'activité et de la connaissance d'une équipe de support informatique. Génie logiciel [cs.SE]. 2011. dumas-00574751

HAL Id: dumas-00574751

<https://dumas.ccsd.cnrs.fr/dumas-00574751>

Submitted on 8 Mar 2011

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.

MEMOIRE

Présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : INFORMATIQUE

OPTION : ISI (INFORMATIQUE, SYSTEMES D'INFORMATION)

Par

LAWSON-BODY Latré Elina épouse AMOYI

CONCEPTION ET DEVELOPPEMENT DE *GECKO*

**LOGICIEL DE GESTION
DE L'ACTIVITE ET DE LA CONNAISSANCE
D'UNE EQUIPE DE SUPPORT INFORMATIQUE**

Soutenu le 14 Janvier 2011

JURY

PRESIDENT : Christophe PICOULEAU

MEMBRES : Bertrand DAVID, Claude GENIER, Christian MASSONNET, Cyril LORENZI

C. Genier
C. GENIER
6/12/2010

REMERCIEMENTS

Je remercie,

Mon Père, pour m'avoir soutenue et aidée pendant le projet et la rédaction du mémoire.

Claude Genier, mon professeur titulaire au Cnam de Saint Genis Pouilly, pour avoir cru en moi et m'avoir motivée tout au long de ce projet. Ses directives et conseils m'ont permis d'aboutir.

Christian Massonnet, mon responsable en entreprise pour avoir encouragé l'initiation et la réalisation du projet, pour son soutien et son implication dans la préparation du mémoire et de la soutenance.

Dany Amoyi, mon époux bien aimé, pour son indéfectible soutien, les défis qu'il m'a lancés pour maintenir ma motivation, et pour s'être occupé de nos enfants pendant la rédaction du mémoire.

Nathalia Martinova, ma mère pour m'avoir soutenue et encouragée.

(Feue) Zinaïda Ivanovna, ma grand-mère, pour ses encouragements et son amour pour moi.

Messieurs Christophe Picouveau et Bertrand David, membres du jury, pour la lecture du mémoire et leur investissement dans cette étape finale du cursus d'Ingénieur du Cnam.

LISTE DES ABREVIATIONS

Abréviations	Signification
DIT	Data Integration Testing
DST	Data System Testing
FTP	File Transfer Protocol
ITQTS	Information Technology Quality Tests Services
MVS	Memory Virtual Storage
RBPM	Risk Based Project Management
RDM	Release Delivery Management
SCP	Secured Copy
SIT	System Integration testing
TM	Test Manager
UAT	Users Acceptance Testing

TRADUCTION DES TERMES ANGLAIS

Terme ou expression en anglais	Traduction française
Conversion	Conversion
Critical	Critique
Data Integration Testing	Tests d'intégration de données
Data System Testing	Test du système de données
Design	Conception
Form	Formulaire
Global Banking and Markets	Secteur bancaire et marchés mondiaux
Helpdesk	Service d'aide à l'utilisateur
High	Elevé
Impact Analysis	Analyse d'impact
Implementation, Configuration and Handover	Implémentation, Configuration et transfert de connaissances
Information Technology Information Library	Bibliothèque de données de la technologie de l'information
Information Technology Quality Tests Services	Services de technologie de l'information et de qualité des tests
Initial Request and Review	Demande initial et révision
Low	Faible
Medium	Moyen
Monitoring	Surveillance
Operations Acceptance Tests	Operations de validation des tests
Private Bank	Banque privée
Program Inception	Initiation du programme
Project Completion Certificate	Certificat d'achèvement de projet
Project terms of engagement	Termes d'engagement du projet
Projects and Tests Environments Support	Support des projets et des environnements de tests
Quality assurance	Assurance Qualité
Quality Center	Centre de qualité
Query	Requête
Refresh	Régénération
Release Delivery Management	Gestion de la mise en production

Terme ou expression en anglais	Traduction française
Report	Rapport, Etat
Requirements Gathering	Collecte des exigences
Risk Based Project Management	Gestion de projet basée sur le risque
System Integration Testing	Tests d'intégration du système
System Integration Testing	Test d'intégration des systèmes
System Testing	Tests des systèmes
Technical Documentation	Documentation technique
Test Activity Completion Certificate	Certificat d'achèvement de l'activité de test
Test Analyst	Analyste de tests
Test Management	Gestion de tests
Test Manager	Gestionnaire de tests
Tests Environments Manager	Gestionnaire des environnements de tests
Trial Conversion	Conversion de processus
Unit Integration Testing	Tests unitaires d'intégration
Unit Testing	Tests unitaires
User Documentation	Documentation des utilisateurs
User Training	Formation de l'utilisateur
Users Acceptance Tests	Tests de validation des utilisateurs
Workflow	Flux de travail

TABLE DES MATIERES

REMERCIEMENTS.....	1
LISTE DES ABREVIATIONS.....	2
TRADUCTION DES TERMES ANGLAIS.....	2
I - INTRODUCTION.....	7
II - PRESENTATION DU PROJET ET DE SON ENVIRONNEMENT.....	9
1 - Présentation du projet GECKO.....	9
2 - Présentation de la société HSBC Private Bank.....	12
3 - Présentation de l'équipe Projects and Tests Environments Support, ses activités et son environnement de travail	13
3.1 L'équipe et son histoire... ..	13
3.2 L'environnement technique de travail	14
3.3 Les tâches de l'équipe.....	17
III - ETAT DES LIEUX.....	21
1 - La méthode actuelle de gestion d'incidents dans l'équipe.....	21
2 - La méthode actuelle de gestion de la connaissance.....	21
3 - Présentation d'un cas concret de suivi d'un incident et de gestion de la connaissance acquise.	23
4 - Analyse du cas présenté plus haut	23
IV - LA DEFINITION DES BESOINS DES UTILISATEURS ET LES CONTRAINTES DU PROJET.....	25
1 - La définition des besoins du projet.....	25
2 - Les contraintes du projet GECKO	31
V - LES ARCHITECTURES FONCTIONNELLE ET TECHNIQUE DU LOGICIEL... 33	
1 - Le schéma Fonctionnel de GECKO	33
1.1 L'identification de l'utilisateur.....	35
1.2 Un menu d'accueil selon le profil connecté	37
1.3 Le descriptif de chacune des fonctionnalités proposées par le menu d'accueil	37
2 - L'architecture technique de GECKO	40
2.1 Descriptions fonctionnelle et technique des tables constituant la base de données GECKO	40
2.2 Le modèle Conceptuel de Données	45
2.3 La méthode de sauvegarde de GECKO	46
2.4 L'hébergement de la Base de Données	47
2.5 La conversion de la documentation existante.....	47
2.6 La gestion des accès utilisateur et des mots de passe	47
2.7 Le couplage de la base de données ACCESS au moteur SQL Server	48
2.8 La sécurité des données	48
VI - DEVELOPPEMENT DE LA SOLUTION RETENUE.....	49
1 - La définition des tables	50
2 - La création des requêtes.....	55
3 - Les outils de saisie et les formulaires	59
4 - Les Etats et la restitution de données	62

5 - Les Macros.....	63
6 - Les graphiques	63
7 - Le démarrage de l'application et l'interface utilisateur.....	64
8 - L'analyse de performance de la base de données et la connexion à SQL SERVER 2005.....	64
9 - La sécurité d'accès à la base de données.....	65
10 - La réplication et la méthode de sauvegarde manuelle de la base de données...	67
VII - LA PHASE DE TESTS DE GECKO.....	69
1 - Les Rôles et Responsabilités des personnes impliquées dans les phases de test du projet	69
2 - La stratégie de test	70
2.1 La phase des tests unitaires	74
2.2 Le Test de validation des utilisateurs (UAT)	77
2.3 La gestion des tests, des anomalies et de l'assurance qualité	80
VIII - MISE EN PRODUCTION DE GECKO.....	89
1 - Les conditions à remplir pour la mise en production de GECKO.....	89
2 - La préparation de la mise en production.....	89
2.1 Le guide de déploiement.....	89
2.2 La stratégie de retour arrière	90
2.3 La définition des droits d'accès des utilisateurs	91
3 - Les phases de mise en production de GECKO.....	91
IX - SUPPORT, MAINTENANCE ET EVOLUTIONS DE GECKO.....	93
1 - Le support fourni aux utilisateurs	93
2 - La restauration de la base de données en cas de panne ou de corruption de données	93
3 - Les développements prévus du logiciel GECKO.....	93
4 - La méthode employée pour la maintenance et l'évolution de GECKO.....	94
X - CONCLUSION.....	97
XI - ANNEXES.....	99
XII - BIBLIOGRAPHIE ET REFERENCES INTERNET.....	105
<i>Liste des figures présentées par le mémoire.....</i>	<i>106</i>
<i>Liste des tableaux présentés par le mémoire.....</i>	<i>107</i>

I - INTRODUCTION

Les outils de travail en entreprise font partie des indicateurs de productivité des équipes. Ils fournissent également des éléments intervenants dans la valorisation des collaborateurs. En particulier, les outils de traçabilité de l'activité sont importants pour la représentation du travail des collaborateurs.

Quelle importance accordent les entreprises à l'évolution des méthodes de travail employées par les collaborateurs ? Les révisions de procédures sont rarement prioritaires face à la pression subie par les équipes de délivrer dans des délais de plus en plus courts.

Ce mémoire présente un projet ayant pour objectif d'optimiser deux méthodes de travail d'une équipe de support des projets informatiques : la gestion de l'activité et la sauvegarde de la connaissance acquise.

Avec comme point de départ l'observation et l'analyse du contexte actuel de travail de l'équipe, le projet détermine ensuite les besoins à l'aide des informations récoltées lors des discussions avec les membres de l'équipe.

A l'issue de ce travail collectif, une solution technique est proposée à savoir la création d'un outil de travail simple et personnalisé pour l'équipe. Cette application permet de tracer l'activité et de gérer la documentation des connaissances acquises.

Le mémoire présente dans un premier temps le cadre, l'équipe et l'entreprise, dans lesquels est réalisé le projet GECKO.

La présentation de l'environnement du projet est suivie d'un descriptif de la méthodologie actuelle de travail de l'équipe, de la définition des besoins et contraintes du projet.

Le mémoire décrira ensuite les solutions fonctionnelle et technique proposées pour répondre aux besoins de l'équipe, ainsi que la phase de développement de l'application GECKO.

En parallèle du développement de l'outil GECKO, une stratégie de test et de formation des utilisateurs sera préparée.

Un guide de déploiement sera rédigé pour la préparation de la livraison de GECKO, incluant la procédure à suivre en cas de problèmes.

Enfin, le plan d'évolution et de maintenance de GECKO viendra clôturer la présentation de ce projet.

II - PRESENTATION DU PROJET ET DE SON ENVIRONNEMENT

1 - Présentation du projet GECKO

Le projet *GECKO* a pour objectif la conception et le développement d'un outil de travail adapté à l'activité de support des tests de projets informatiques.

Le projet *GECKO* se déroule au sein de la société HSBC Private Bank de Genève (Suisse), dans l'équipe Projects and Tests Environments Support. Cette équipe fournit un support technique et organisationnel aux projets informatiques durant leurs différentes phases de test.

GECKO répondra à un besoin de gestion et traçabilité des incidents traités par l'équipe, de sauvegarde de la connaissance acquise au fil des interventions, d'amélioration de la communication des informations et événements se déroulant dans les environnements de tests.

La nécessité du développement d'un tel outil est apparue face à la croissance exponentielle de l'activité de l'équipe Projects and Tests Environments Support et à l'exigence de plus en plus poussée des projets informatiques en termes de Support.

La définition des étapes du projet se base sur la méthode de gestion de projet RBPM (Risk Based Project Management), définie comme standard des projets informatiques de la Banque. Toutes les étapes du modèle RBPM ne seront pas suivies du fait de la taille et des contraintes du projet *GECKO*. La figure ci-dessous présente la méthode RBPM.

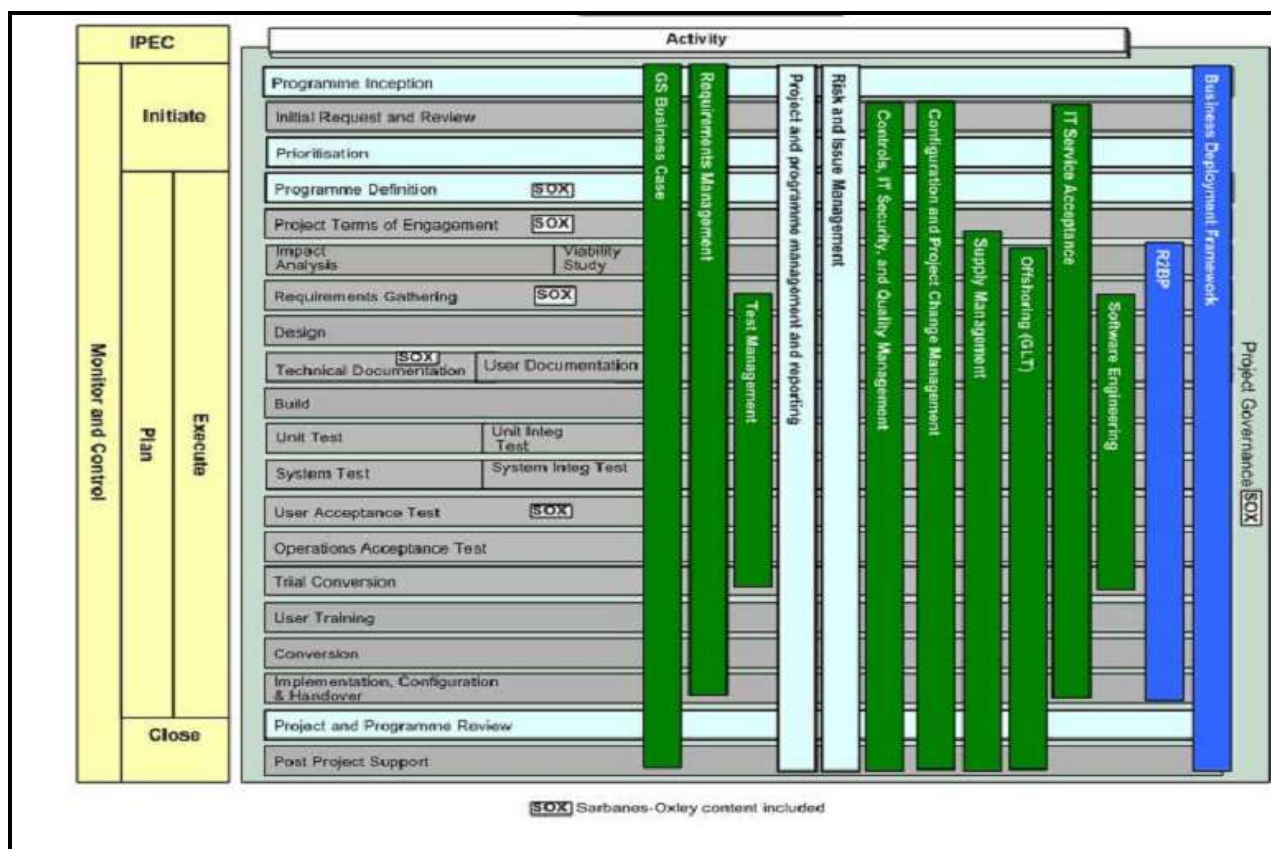


Figure II.I.1 : Modèle de gestion de projet RBPM :Risk Based Project Management

Le besoin d'une solution exprimé par le responsable de l'équipe a été la première étape du projet. Une étude de l'état des lieux et l'expression détaillée des besoins de l'équipe seront établies par la suite. A l'aide des contraintes délimitant le projet, des solutions fonctionnelle et technique seront proposées au responsable de l'équipe pour validation. Une fois ces propositions validées, le développement du logiciel GECKO pourra être lancé. Une formation des utilisateurs est prévue, avant la mise en production. Le schéma ci-dessous propose un aperçu des étapes qui seront suivies par le projet :

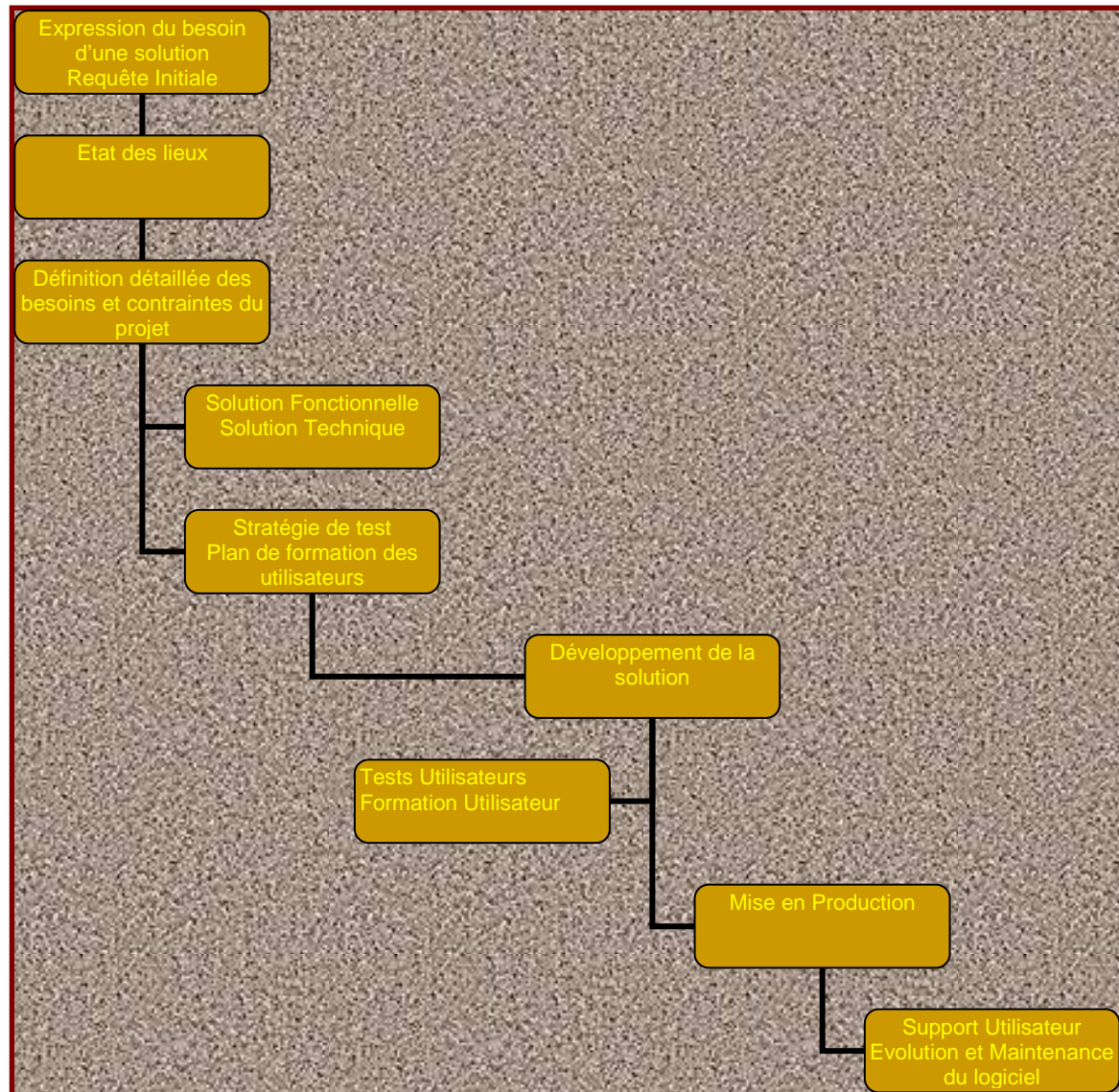


Figure II.I.II : Etapes du projet GECKO

En supplément de la méthode RBPM, la norme ITIL V3 (Information Technology Information Library Version 3) sera également utilisée lors du projet afin d'enrichir les étapes du cycle de vie du logiciel GECKO et de respecter l'aspect **service fourni au client**.

ITIL est une norme qui propose un modèle de gestion des relations entre le département informatique et ses clients. C'est un recueil de méthodes d'optimisation de la satisfaction des besoins des clients par les services informatiques, d'amélioration des processus de travail entre le département informatique et ses clients.

La notion de service fourni au client est la base de la norme ITIL. Le service fourni au client est présenté comme une prestation qui doit être utile et garantie, une prestation qui a de la valeur pour le client.

ITIL propose les étapes, présentées dans la figure ci-dessous, pour définir le cycle de vie d'un service.

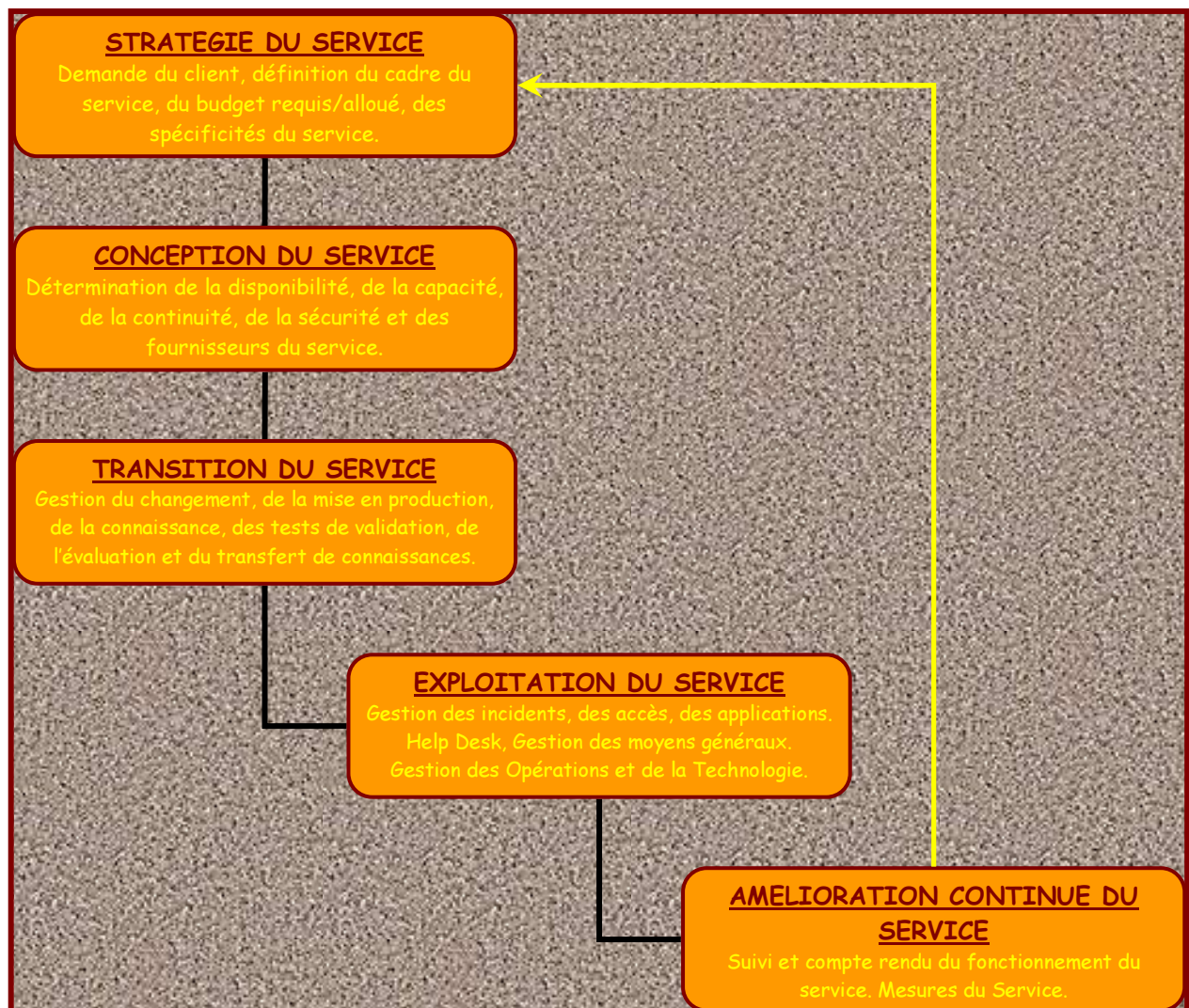


Figure II.I.III : Etapes du cycle de vie du Service proposées par ITIL

Le projet se place dans une optique de livraison de service à son client : l'équipe Projects and Tests environments Support. La norme ITIL V3 fournit les outils nécessaires à l'assurance qualité du logiciel GECKO. Un des objectifs d'ITIL est également de déterminer dès la phase de stratégie si un projet est faisable ou pas.

L'équipe du projet GECKO est composée de deux personnes de l'équipe Projects and Tests Environments Support. Le responsable de l'équipe qui est le sponsor du projet et le collaborateur en charge d'exécuter toutes les tâches inhérentes aux étapes du projet mentionnées ci-dessus. Une réunion de suivi du projet est organisée deux fois par mois.

2 - Présentation de la société HSBC Private Bank

Le groupe HSBC, dont le quartier général est situé à Londres, est l'une des plus grandes institutions fournissant des services bancaires et financiers à travers le monde.

Son réseau international compte 8000 locations dans 88 pays et territoires en Europe, à Hong Kong, en Asie Pacifique, au Moyen Orient, en Afrique, en Amérique du nord et du sud.

Le groupe HSBC fut fondé à Hong Kong en 1865 pour financer le commerce entre la Chine, l'Europe et les Etats-Unis. Dès lors, le groupe s'est agrandi grâce aux rachats et à une croissance internationale.

Les actifs du groupe HSBC représentaient 2364 milliards de dollars au 31 Décembre 2009, dont un profit brut de 13,3 milliards de dollars.

Le groupe HSBC fournit une palette considérable de services financiers à plus de 100 millions de clients à travers quatre type d'activités : les services financiers personnels, la banque commerciale, le Global Banking and Markets et la banque privée.

Les entités du groupe HSBC sont HSBC Private Bank (Grande Bretagne) Limited (précédemment Samuel Montagu & Co Limited), fondée en 1853, HSBC Trinkaus & Burkhardt KG (fondée en 1785), HSBC Guyerzeller Bank AG (fondée en 1866), HSBC Bank USA (ancienne Republic National Bank of New York fondée en 1966) et le Crédit Commercial de France (CCF fondé en 1894).

HSBC Private Bank (anciennement HSBC Republic), fut fondée le 31 Décembre 1999, lors de l'acquisition par le groupe HSBC des banques Republic New York Corporation et Safra Republic Holdings, filiales de la Republic National Bank of New York. Ses deux banques fondées en 1966 dans une tradition bancaire datant de l'empire Ottoman, étaient spécialisées dans le domaine de la banque privée.

Depuis cette acquisition, l'activité de HSBC Private Bank n'a cessé d'augmenter grâce à des fusions, au développement de nouveaux produits, à l'intégration de nouveaux marchés, à l'expertise et au professionnalisme de ses collaborateurs.

Le nom commercial HSBC Private Bank fut adopté le 1er Janvier 2004. L'utilisation du label 'HSBC Private Bank' fait référence l'activité internationale de Banque privée du groupe HSBC.

HSBC Private Bank est la dénomination commerciale de l'activité de gestion privée des principales filiales de banque privée du Groupe HSBC à travers le monde. Elle offre des services financiers à une clientèle fortunée dans 91 implantations réparties dans 39 pays en Amérique du Nord et du Sud, en Asie-Pacifique, en Europe, au Moyen-Orient et en Afrique. Ses actifs sous gestion représentent 460 milliards de dollars au 31 décembre 2009.

Implantée à Genève en Suisse, HSBC Private Bank compte aujourd'hui 2000 employés, dont 250 dans le département informatique.

3 - Présentation de l'équipe Projects and Tests Environments Support, ses activités et son environnement de travail

3.1 L'équipe et son histoire...

L'équipe Projects and Tests Environments Support, fait partie du département ITQTS, InformationTechnology Quality Test Services. Ce département regroupe les activités de gestion, de support et d'assurance qualité des tests effectués par les projets informatiques de la banque dans toute l'Europe.

Le département ITQTS est composé de trois équipes : Projects and Tests Environments Support, Quality Assurance (Assurance Qualité) et Test Management (Gestion des tests).

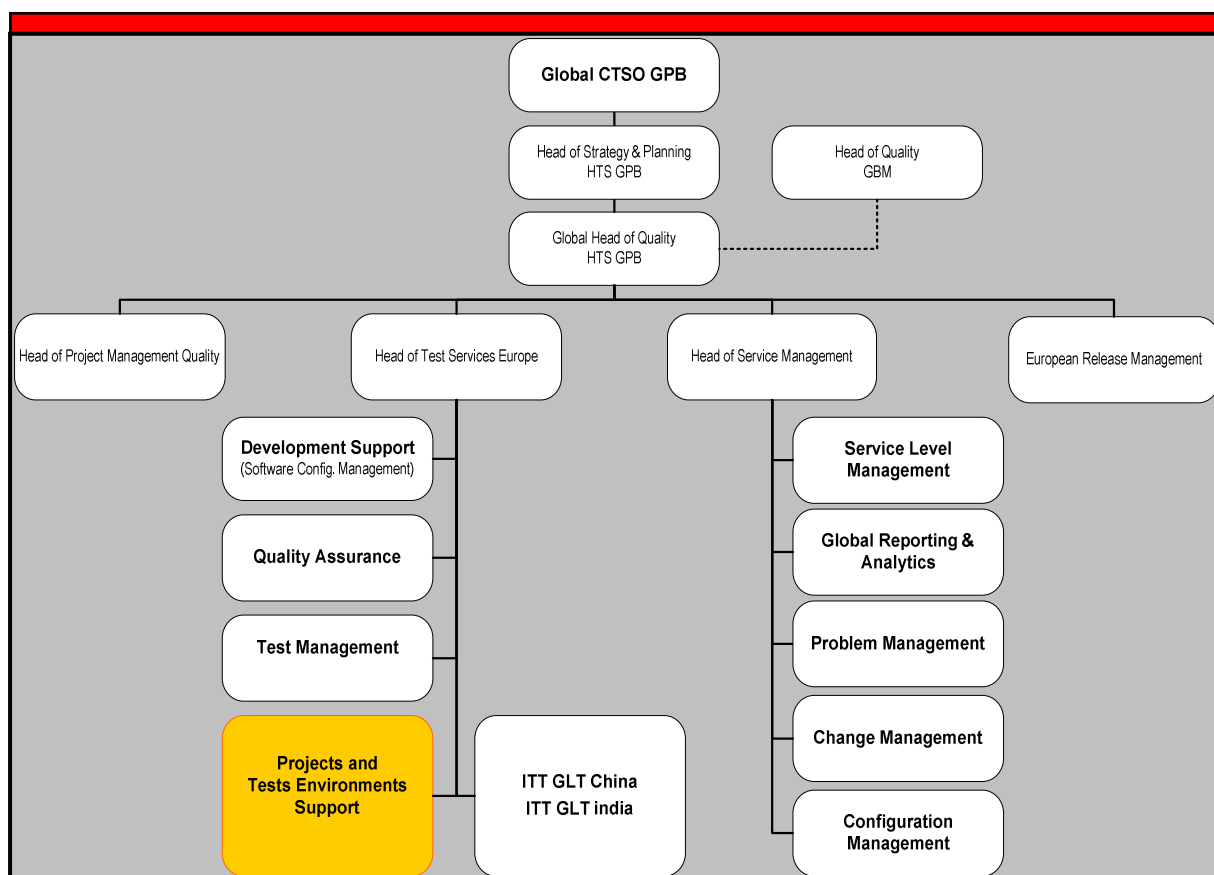


Figure II.III.I Organigramme montrant la structure du département ITQTS dont fait partie l'équipe

L'équipe Projects and Tests Environments Support est composée de cinq personnes : quatre employés et un responsable d'équipe. Le domaine de compétences techniques et fonctionnelles des membres couvre la plupart des applications présentes dans les environnements de tests des projets informatiques.

Constituée en 2003 d'une personne, l'équipe a commencé ses activités avec la gestion des environnements de test des projets, la coordination des mises à jours des tables de bases de données dans les environnements de tests et de production et l'administration des accès utilisateurs aux environnements de test.

L'équipe a ensuite connu une très forte croissance en cinq ans tant au niveau de ses membres qu'au niveau de la connaissance et des activités, étendant ainsi son niveau d'expertise et d'intervention.

Elle compte aujourd'hui cinq membres et est prévue d'atteindre huit personnes à la fin de l'année 2011. Toute la connaissance spécifique aux applications et à leurs interfaces a été acquise grâce au support fourni aux projets durant les sept dernières années.

L'expertise et la compétence de l'équipe viennent des connaissances techniques que chaque membre a su mettre en commun avec l'apprentissage du fonctionnement des logiciels de la banque. Le niveau de diagnostic s'en est trouvé amélioré et le temps de résolution des incidents diminué.

La maturité et les compétences de l'équipe sur les problématiques de toutes les phases de tests et des environnements de tests des projets, ont permis de faire évoluer le nom du poste de ses membres de l'équipe de Test Environments Manager à Test Analyst. C'est une reconnaissance de l'expertise et de la qualité de support apportées par l'équipe aux projets.

3.2 L'environnement technique de travail

L'équipe gère seize environnements de tests informatiques, répartis en quatre niveaux. Chaque environnement de test contient en moyenne vingt applications. Le niveau de sensibilité des environnements couvre le développement, jusqu'aux tests fonctionnels des utilisateurs. Les départements utilisant les applications bancaires sont en général les sponsors des projets auxquels l'équipe fournit un support durant les phases de test.

Les projets transitent par quatre environnements de tests durant leur cycle de vie : DEV, DST, DIT et UAT.

Le DEV : C'est l'environnement utilisé pour le développement des programmes et pour le codage. Il existe un environnement de développement pour chaque application.

Le DST (Data System Testing) : C'est l'environnement utilisé pour effectuer les tests unitaires des programmes en cours de développement.

Le DIT (Data Integration Testing) ou SIT (System Integration Data): Cet environnement de test est utilisé pour intégrer le ou les nouveau programmes à l'application et valider l'absence de conflit avec les programmes existants. Puis l'application, ainsi mise à jour, est à son tour intégrée à l'ensemble des applications existantes. Il est ainsi possible de valider leur cohabitation et le flux d'échanges inter applications.

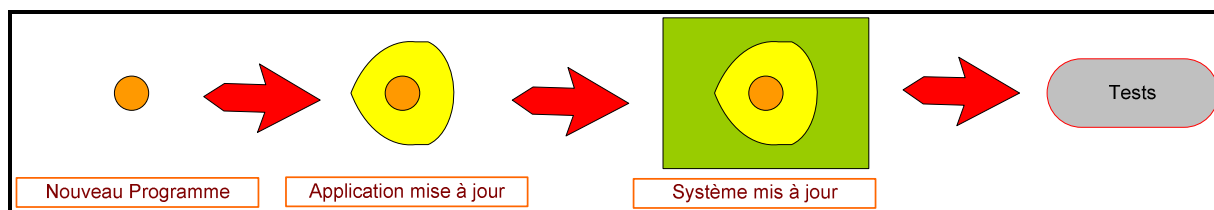


Figure II.III.II : Fonctionnement du DIT

L'UAT (User Acceptance Testing) : Cet environnement est utilisé pour les tests de validation effectués par l'utilisateur final de l'application.

Le tableau ci-dessous présente les environnements de tests supportés par l'équipe, avec le nombre d'applications contenues par chaque environnement.

Tableau 2.3.3 : Environnements de test et nombre d'applications par environnement

Nom de l'environnement de test	Nombre d'applications
DEV	30 applications
DST 1	10 applications
DST 2	3 applications
DIT 1	38 applications
DIT 2	1 application
DIT 3	10 applications
UAT 1	14 applications
UAT 2	50 applications
UAT 3	6 applications
UAT 4	1 application
UAT 5	3 applications
UAT 6	5 applications
UAT 7	14 applications
TRAINING	5 applications
REG 1 (utilisé pour les tests de régression)	2 applications
REG 2 (utilisé pour les tests de régression)	1 application

La structure du réseau des applications suit un modèle en forme de satellite. Les applications sont alimentées par un système central, le mainframe.

Le schéma ci-dessous représente le mainframe relié à des applications hébergées par différents systèmes d'exploitation.

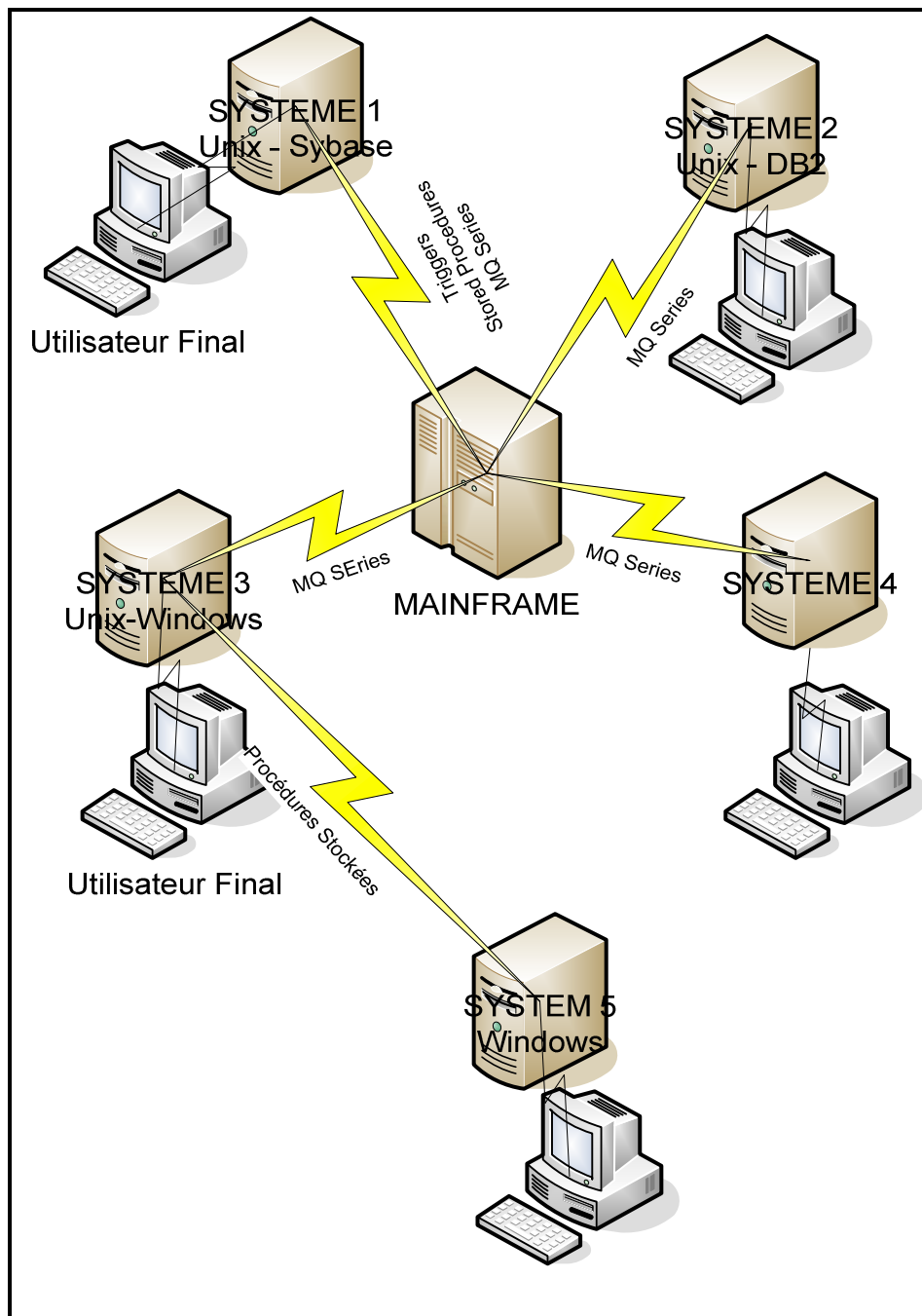


Figure II.III.IV : Mainframe relié aux applications

Le mainframe est installé sur le système d'exploitation MVS (Memory Virtual Storage). Ce système d'exploitation fournit et maintenu par IBM existe depuis 1974. Il est utilisé par les entreprises possédant de grandes bases de données. Par exemples les hôpitaux, les banques, les compagnies assurances.

Les programmes du mainframe sont écrits en langage Cobol, Assembleur et Ideal Datacom. La base de données installée sur le mainframe est de type Datacom.

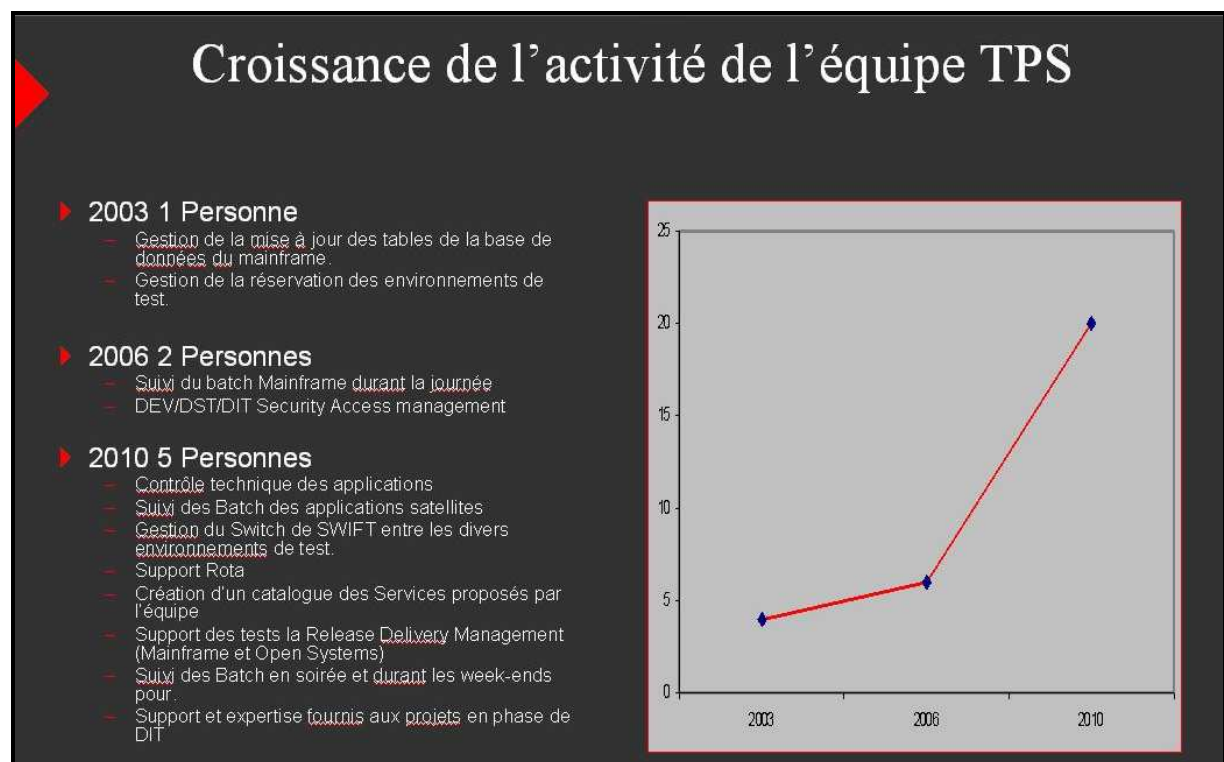
Les applications satellites du mainframe sont installées sur UNIX et WINDOWS. Leurs bases de données sont de type ORACLE, SYBASE, SQL SERVER, DB2, DATACOM, MySQL.

L'échange de flux entre les interfaces des applications se fait via des liens MQ Series, des procédures stockées, des scripts et par batch (chaîne de programmes tournant en fin de journée comptable). Le batch utilise les protocoles de transfert FTP (File Transfer Protocol) et SCP (Secure Copy).

3.3 Les tâches de l'équipe

Le nombre de tâches exécutées par l'équipe a connu une forte croissance aux cours des sept dernières années. Cette croissance continue chaque année en adéquation avec les besoins des projets et les normes de la banque.

La courbe ci après montre l'évolution du volume d'activité de l'équipe.



En ordonnées : Le volume des tâches exécutées par l'équipe.

En abscisse : Les années depuis la création de l'équipe.

Nota Bene : Les activités de ce graphique s'accumulent au fil des années.

Figure II.III.V : Courbe de la croissance des activités de l'équipe Projects and Tests Environments Support

L'équipe Projects and Tests Environments Support, a pour mission de livrer des environnements informatiques opérationnels et à même de supporter les tests prévus par les projets. Tout au long des tests, l'équipe est chargée de maintenir ces environnements opérationnels, de les adapter si besoin est, de fournir l'assistance nécessaire en cas de problèmes techniques.

Le processus de préparation et de livraison des environnements se fait à partir d'un travail en amont avec le chef de projet.

Le chef de projet remplit un formulaire de réservation de l'environnement de test qui l'intéresse, en précisant les dates, les applications nécessaires aux tests, le type de données, l'accord ou pas de partager l'environnement avec un autre projet, la nécessité d'un batch....

A partir de ces pré-requis, l'équipe prépare l'environnement. Elle coordonne l'installation ou la mise à jour de programmes, la construction d'un nouveau composant. Elle met au courant les autres projets travaillant dans l'environnement, gère les éventuels conflits en proposant des solutions en fonction des besoins de chaque projet, organise des réunions pour mettre les projets d'accord. Afin d'être pro actifs pour le support du projet, les Tests Analysts prennent connaissance de l'architecture technique du projet avant le début des tests.

L'équipe Projects and Tests Environments Support est garante de la disponibilité et de la fiabilité des environnements de tests fournis aux projets. Elle est également le principal support durant les phases de test.

La coordination de l'installation des nouvelles tables de bases de données sur tous les environnements de tests et de production est également gérée par l'équipe. Cette activité consiste à établir un calendrier d'installation pour chaque environnement en accord avec les chefs de projets et les équipes techniques. Une fois le calendrier validé, l'équipe est en charge de la communication des statuts aux projets au fil des installations. En cas de conflits ou d'événements inattendus empêchant le bon déroulement du calendrier, l'équipe se charge de trouver des solutions et de garantir la mise à jour de tous les environnements.

Chaque membre de l'équipe est rattaché à un ou plusieurs projets dès leur phase d'initiation. Le travail de Test Analyst implique la participation à toutes les réunions techniques du projet, le suivi de la préparation ou de la construction de l'environnement de test du projet, l'administration des droits d'accès des utilisateurs aux applications présentes dans les environnements de tests, la validation des solutions techniques proposées par le projet et la validation de la stratégie de test du projet.

L'organigramme ci-dessous montre la place du Test Analyst dans une équipe de projet.

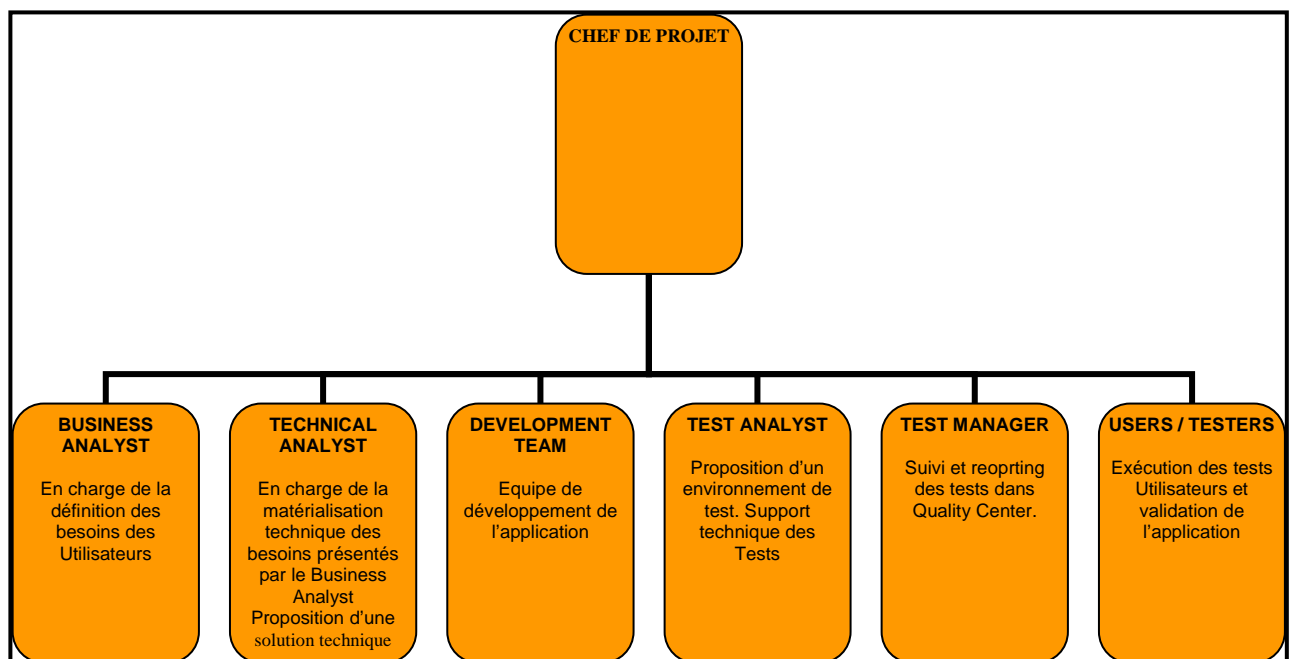


Figure II.III.IV : Structure de l'équipe projet incluant le Test Analyst

Une fois la phase des tests démarrée par le projet, le Test Analyst apporte une expertise pour l'investigation et la résolution de problèmes techniques et parfois fonctionnels que rencontre le projet.

L'équipe Projects and Tests Environments Supports joue un rôle important dans la préparation et le déroulement de la Release Delivery Management (RDM). Il s'agit d'un processus de gestion de projets qui consiste à regrouper tous les projets planifiés dans l'année et à leur imposer quatre dates de mise en production annuelles.

Ceci permet de tester toutes les modifications de programmes ensemble, de corriger les problèmes relatifs à la cohabitation et de livrer des applications plus stables et aptes à la cohabitation. Les tâches de l'équipe dans la Release Delivery sont :

- La préparation de tous les environnements de test dédiés à la RDM :
 - La validation des environnements de test après les refresh (rapatriement de données de production en environnement de test).
 - L'application des paramètres spécifiques aux environnements de tests. (Exemple : les imprimantes, les noms de Channel, les proxys etc.....).
 - La vérification de tous les liens techniques entre les applications impliquées dans les tests de la RDM.
 - La validation fonctionnelle des liens entre les applications. Cette tâche consiste à effectuer une opération comme elle le serait en production par l'utilisateur. Par exemple un achat de titre, un virement.
- La coordination de la gestion des droits d'accès utilisateurs entre le département de sécurité et les utilisateurs.
- Le support en cas de problèmes techniques liés aux environnements de tests. L'équipe se doit d'être réactive afin que les tests ne soient pas impactés par un délai important.
- L'exécution et le monitoring de la chaîne de programmes de fin de journée production (batch).
- Le diagnostic et le support niveau un et deux des problèmes rencontrés lors du batch. Le support de niveau un est l'investigation sur l'origine du problème. Le support niveau deux est la résolution d'un problème technique et lié à l'environnement de test et non à l'application.
- L'enregistrement et l'assignation des anomalies relatives aux tests dans l'outil Hewlett Packard Quality Center.

Le poste de Test Analyst requiert une parfaite connaissance technique des applications de la banque et de leur interfaçage les une avec les autres. Le Test Analyst doit également être au courant de tous les projets en cours, d'être en possession de tous les éléments relatifs aux tests afin d'être réactif sur les problèmes, les interrogations soulevées par les projets, de savoir mesurer et annoncer l'impact des événements. La compréhension du métier bancaire est aussi indispensable, par exemple pour comprendre et situer un problème, lorsqu'il est directement adressé par un utilisateur.

III - ETAT DES LIEUX

1 - La méthode actuelle de gestion d'incidents dans l'équipe

Les demandes de traitement d'incident sont adressées à l'équipe par le biais de la messagerie électronique, du téléphone, de la messagerie instantanée ou d'une conversation avec la personne soumettant le problème ou réclamant un support.

Les incidents signalés par messagerie électronique :

Lorsqu'une demande d'intervention arrive par email, le collaborateur qui la traite, classe l'email dans son répertoire personnel. L'email n'est alors plus visible dans la messagerie de l'équipe.

Les incidents signalés par téléphone, messagerie instantanée ou verbalement :

Ces incidents sont notés par le collaborateur sur un cahier, et traités sans qu'il n'y ait aucune trace pour le reste de l'équipe ni de la demande initiale, ni du statut, ni de la résolution.

La difficulté majeure réside donc dans l'impossibilité de suivre un incident ou un dossier par voie électronique, le seul canal de transmission reste la communication orale d'informations entre les membres de l'équipe.

Il existe un logiciel utilisé par l'équipe du Helpdesk permettant de gérer les incidents et demandes d'intervention de tous les utilisateurs de la société. Bien qu'ayant l'avantage de pouvoir être utilisé par tous les services du département informatique, ce logiciel reste lent et lourd du fait des nombreuses informations qui doivent être obligatoirement saisies lors de la déclaration d'un incident. Ces informations n'étant pas toutes utiles dans la gestion des incidents liés aux tests des projets, l'équipe perdrait beaucoup de temps dans la saisie lors de la création d'un dossier lié à un incident. L'utilisation en est donc assez rébarbative.

2 - La méthode actuelle de gestion de la connaissance.

La connaissance de l'équipe s'est construite et améliorée au fil des interventions et du support fourni aux projets informatiques durant les phases de test. Des étapes parcourues lors de la résolution d'un incident, les membres de l'équipe essaient d'écrire des procédures lorsque le temps le leur permet. Dans le cas contraire, les informations récoltées durant le diagnostic, l'investigation et la résolution de l'incident restent dans un cahier ou sur un papier. Les autres membres de l'équipe n'y ayant pas eu accès, ils devront reprendre tout le processus de recherche d'informations si le même incident se reproduisait. Il existe également le risque que même la personne ayant gardé ses notes sur un papier ne puisse se relire ou manquer d'informations dans ses propres notes.

La figure, ci après montre un exemple de notes prises par un collaborateur. Une partie de ces notes devrait être documentée afin de sauvegarder la connaissance. Par manque de temps, elle ne le sera pas. Il n'existe aucun moyen fiable de référencer ces informations. Le collaborateur devra faire appel à sa mémoire pour les retrouver.

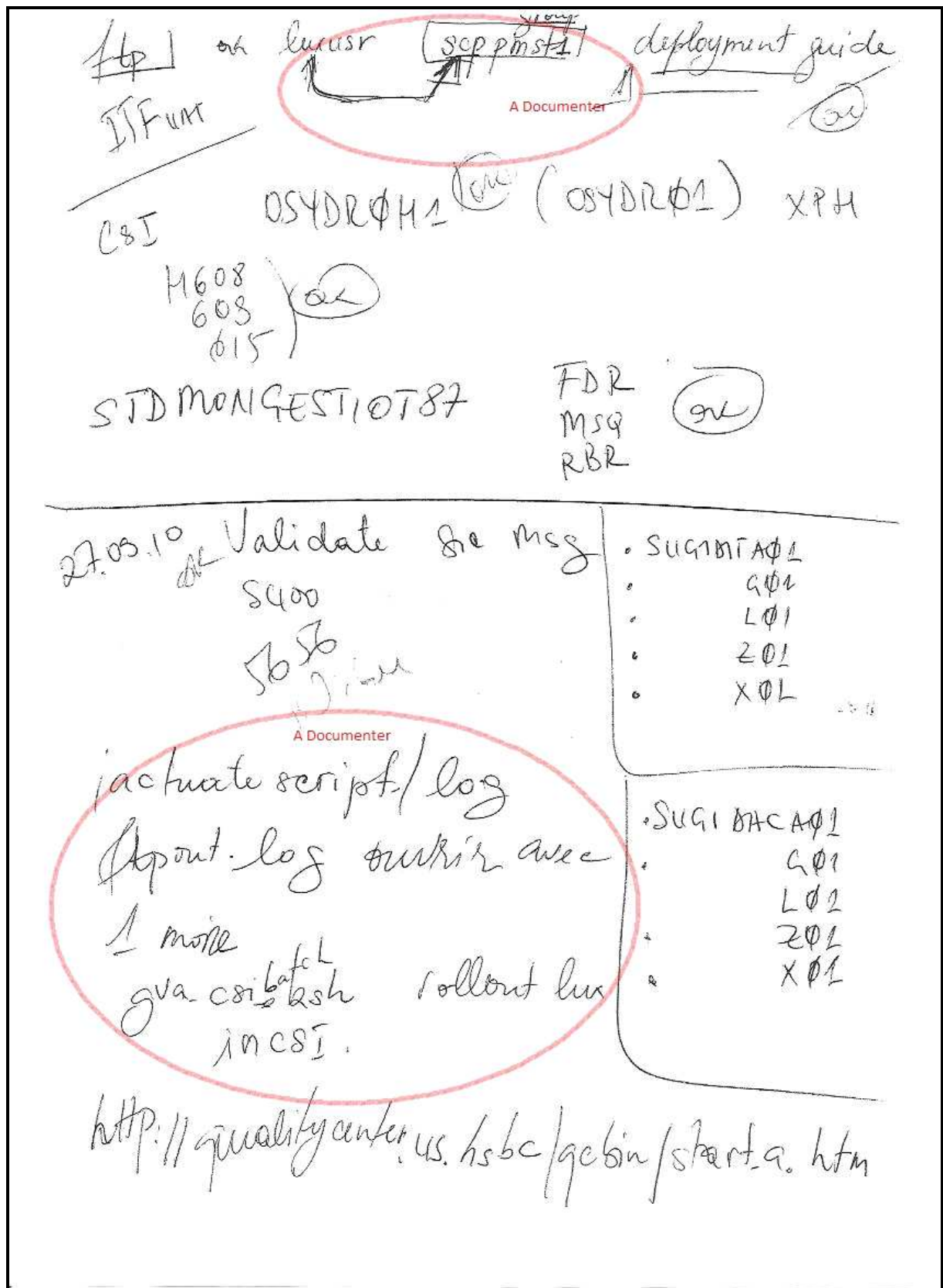


Figure III.II.1 : Un exemple de notes prises par un collaborateur

3 - Présentation d'un cas concret de suivi d'un incident et de gestion de la connaissance acquise.

A 13 heures, une déclaration d'incident parvient à l'équipe par téléphone. Le chef d'un projet appelle le membre A de l'équipe avec lequel il a de bonnes relations et lui expose le problème rencontré : les messages envoyés depuis le mainframe ne parviennent pas à l'application destinataire.

A prend note, sur son cahier, des informations relatives à l'incident (type de message, fréquence d'envoi, dernière heure d'envoi, référence du message, environnement de test concerné, modifications de programmes effectuées par le projet ...), puis promet de rappeler le chef de projet dans un délai de deux heures.

A commence à investiguer, il effectue des tests techniques afin de cerner le problème, se rend compte que ce problème lui est inconnu et qu'il ne dispose pas de connaissances suffisantes. A rassemble des informations en discutant avec d'autres services et les note sur son cahier. Afin de poser un diagnostic définitif, A demande au chef de projet d'effectuer un nouveau test pour tracer le message. En attendant que le chef de projet contacte ses utilisateurs et génère le message, le temps passe et A travaille sur d'autres incidents.

Lorsque le chef de projet rappelle pour signaler que le message est sur le point d'être envoyé, A a fini sa journée et assure le chef de projet que son collègue B reprend le suivi. Malheureusement B étant occupé sur un autre incident, A n'arrive pas à lui communiquer toutes les informations avant de rentrer chez lui.

Le chef de projet appelle B pour lui notifier l'envoi du message, mais B est incapable de déterminer la meilleure approche du problème. Il demande au chef de projet les mêmes informations récoltées par A plus tôt et promet de rappeler le chef de projet une fois qu'il se sera familiarisé avec le problème. B contacte d'autres services et pose les mêmes questions qu'A aux mêmes personnes. La journée tire à sa fin et B avant de rentrer chez lui laisse un message à A, l'informant qu'il devrait reprendre le dossier où il l'avait laissé.

Le lendemain, A arrive tôt au bureau et doit attendre l'arrivée du chef de projet et des utilisateurs avant de reprendre la résolution de l'incident. Le message est enfin régénéré à 10 heures. Le diagnostic définitif est posé. A a besoin de l'intervention d'une autre équipe pour corriger le problème. L'aide de cette équipe permet la résolution définitive du problème à 12 heures. A cet instant un travail urgent survient et A s'y attelle aussitôt...

4 - Analyse du cas présenté plus haut

L'analyse du cas présenté plus haut, montre des faiblesses sur les points suivants :

La saisie des informations

Les informations recueillies lors de la déclaration de l'incident n'ont pas été accessibles à l'équipe. Rendant plus difficile la reprise du suivi par un autre membre de l'équipe.

L'acquisition de la connaissance

Les informations recueillies par A lors de l'investigation sont restées notées sur un cahier et n'ont pas été accessibles à l'équipe.

La transmission des informations

Les seules informations transmises, l'ont été de manière orale et rapidement alors que *A* finissait sa journée de travail, *B* était occupé par autre chose et n'avait pas pensé à prendre de notes.

Le temps de résolution de l'incident

23 heures ont été nécessaires à la résolution de l'incident. Ce délai se décompose comme suit :

1. Déclaration de l'incident.
2. Demande d'informations auprès du chef de projet par *A*.
3. Tests techniques.
4. Recherche d'informations auprès d'autres services par *A*.
5. Demande d'un nouveau test.
6. Attente du test.
7. Demande d'informations auprès du chef de projet par *B*.
8. Recherche d'informations auprès d'autres services par *B*.
9. Attente du test.
10. Intervention d'une autre équipe.
11. Résolution de l'incident.

Les étapes 2, et 4 et 6 du processus ont été répétées.

Quelles sont les méthodes de communication d'informations à mettre en place afin de réduire le temps de résolution d'un incident par l'équipe ?

IV - LA DEFINITION DES BESOINS DES UTILISATEURS ET LES CONTRAINTES DU PROJET

1 - La définition des besoins du projet

L'analyse de la méthode actuelle de gestion des incidents, les plaintes reçues par rapport aux délais de résolution des incidents, l'analyse de la communication au sein de l'équipe ont montré un besoin d'une méthode de travail commune de gestion de l'activité de support aux projets et de transmission de l'information.

Le besoin recensé concerne également une méthode de traçabilité des activités de l'équipe par le responsable de l'équipe afin de fournir les statistiques des tâches exécutées. L'accès à de telles informations serait d'une grande aide pour l'organisation de l'équipe, les formations ou les justifications de demande de budget.

Les deux critères qui seront utilisés pour effectuer l'étude des besoins de l'équipe sont : « Quelles sont les raisons pour lesquelles un outil de travail spécifique à l'équipe doit être développé ? » et « Quelles sont les fonctionnalités que l'on attend d'un tel outil ? »

Le développement d'un logiciel pour l'équipe est justifié par les points suivants:

- Un besoin d'amélioration de la transmission d'informations au sein de l'équipe.
- Un besoin d'une méthode électronique de suivi des incidents gérés par l'équipe.
- Un besoin d'une méthode électronique de suivi des phases de test des projets.
- Un besoin d'avoir des statistiques sur le volume de l'activité de l'équipe.
- La sauvegarde de la connaissance acquise.

Les caractéristiques et fonctionnalités qui devront être privilégiées lors de la conception de l'outil de travail:

Afin de déterminer la méthode de travail la plus appropriée, une fiche de sondage a été circulée au sein de l'équipe, des statistiques ont été menées sur les délais de résolution de chaque incident et chaque étape de la résolution a été analysée.

La nécessité d'utiliser un logiciel dans la gestion des incidents est apparue comme étant la meilleure solution compte tenu des paramètres cités plus haut. Ce logiciel sera nommé GECKO.

La définition des spécifications techniques et fonctionnalités de GECKO seront déterminées par la liste des tâches qu'effectue l'équipe ainsi que la manière dont sont gérées et transmises les informations. Un facteur très important rentrera en ligne de compte : le temps. Le temps de saisie, de l'utilisateur doit être minimal au possible afin de réduire le sentiment de corvée de l'utilisateur. Le temps de réponse du logiciel doit également être performant. Le sondage réalisé auprès des membres de l'équipe a montré les critères suivant comme étant indispensables:

1. La convivialité
2. La fiabilité
3. Le temps de réponse
4. La facilité d'utilisation
5. Un faible nombre de champs à remplir
6. Le temps de restitution des données
7. La clarté des informations restituées

Les fiches de sondage qui ont été circulées dans l'équipe sont les suivantes :

Tableau 4.1.1 : Fiche de Sondage Numéro Un

SONDAGE GECKO #1 Interface Logiciel					
PROFIL :					
<p>Quelles sont caractéristiques importantes d'un logiciel de gestion des incidents traités par l'équipe ?</p>					
<p>Quelle priorité attribuerez-vous aux caractéristiques ci dessous d'un logiciel de gestion d'incidents ?</p>					
<p>* 1 étant la priorité la plus haute et 5 la plus faible</p>					
LOGICIEL	1	2	3	4	5
CONVIVIALITE					
FIABILITE					
TEMPS DE REPONSE					
FACILITE D'UTILISATION					
CHAMPS	1	2	3	4	5
TEMPS DE REMPLISSAGE					
NOMBRE DE CHAMPS					
CLARTE DES INTITULES					
RESTITUTION DES DONNEES	1	2	3	4	5
TEMPS					
CLARTE DES INFORMATIONS RESTITUEES					

Tableau 4.1.2 : Fiche de Sondage Numéro Deux

SONDAGE GECKO #2 Qualité et Performance					
<div style="border: 1px solid black; padding: 2px; margin-bottom: 10px;">PROFIL :</div> <p style="text-align: center;">Quelle note attribueriez-vous à la performance de l'équipe pour les activités suivantes ? * 1 étant la note la plus haute et 5 la plus faible</p>					
LA GESTION D'INCIDENTS	1	2	3	4	5
Gestion des réclamations					
Suivi des dossiers					
Temps de résolution					
Capitalisation des informations acquises					
Commentaires <div style="border: 1px solid black; height: 30px; width: 100%; margin-top: 5px;"></div>					
LA COMMUNICATION DES INFORMATIONS	1	2	3	4	5
TEMPS DE TRANSMISSION DES INFOS					
FIABILITE DES INFORMATIONS TRANSMISES					
Commentaires <div style="border: 1px solid black; height: 30px; width: 100%; margin-top: 5px;"></div>					
LA GESTION DE LA DOCUMENTATION	1	2	3	4	5
TEMPS DE REDACTION D'UNE DOCUMENTATION					
CLARTE DE LA DOCUMENTATION					
Commentaires <div style="border: 1px solid black; height: 30px; width: 100%; margin-top: 5px;"></div>					

Les résultats du sondage :

Le sondage a été mené auprès des 5 membres de l'équipe. Chaque colonne contient le nombre de personnes ayant attribué la note mentionnée en tête de colonne.

Tableau 4.1.3 : Résultat du sondage Numéro un, les caractéristiques du logiciel à concevoir

RESULTAT SONDAGE GECKO #1					
LOGICIEL	1	2	3	4	5
CONVIVIALITE					
		3	2		
FIABILITE	5				
TEMPS DE REPONSE	5				
FACILITE D'UTILISATION	5				
CHAMPS	1	2	3	4	5
TEMPS	5				
NOMBRE	5				
CLARTE	5				
RESTITUTION DES DONNEES	1	2	3	4	5
TEMPS					
		4	1		
CLARTE	5				

Tableau 4.1.4 : Résultat du sondage Numéro deux, performance de l'équipe et qualité de l'activité

RESULTAT SONDAGE GECKO #2					
<div style="border: 1px solid black; padding: 2px; display: inline-block; color: #c00000; font-weight: bold;">PROFIL :</div>					
LA GESTION D'INCIDENTS	1	2	3	4	5
Gestion des réclamations	1	4			
Suivi des dossiers		2	3		
Temps de résolution				5	
Capitalisation des informations acquises				5	
<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Commentaires</div> <div style="border: 1px solid black; padding: 5px; width: 80%;">La personne qui suit le dossier est la seule à en connaître le statut</div> </div>					
LA COMMUNICATION DES INFORMATIONS	1	2	3	4	5
TEMPS DE TRANSMISSION DES INFOS				5	
FIABILITE DES INFORMATIONS TRANSMISES		3	2		
<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Commentaires</div> <div style="border: 1px solid black; padding: 5px; width: 80%;">Il faut toujours écrire un long mail explicatif en cas de transmission du dossier à un collègue</div> </div>					
LA GESTION DE LA DOCUMENTATION	1	2	3	4	5
TEMPS DE REDACTION D'UNE DOCUMENTATION				1	4
CLARTE DE LA DOCUMENTATION				5	
<div style="display: flex; justify-content: space-between;"> <div style="width: 20%;">Commentaires</div> <div style="border: 1px solid black; padding: 5px; width: 80%;">Il y a souvent plusieurs fichiers sur le même thème. Il est difficile de trouver du premier coup ce que l'on cherche</div> </div>					

Le sondage a en outre permis de recueillir les commentaires suivants :

De la part du responsable de l'équipe :

- En l'absence d'un collaborateur, il est difficile de trouver rapidement les dossiers sur lesquels il travaille, ou de connaître leur état d'avancement
- Pour valider une action ou prendre une décision sur un incident, il n'y a aucun accès à l'historique de son suivi.
- Il est impossible d'obtenir des statistiques de l'activité de l'équipe. Par membre, par application, sur une période donnée ...
- Les procédures et la documentation ne sont pas toutes à jour. C'est un élément important pour faire face aux audits de sécurité.
- Il n'y a pas de possibilité de connaître tous les événements du jour dans les environnements de test. Cette information n'est centralisée nulle part.
- Il n'y a pas d'outil pour suivre l'introduction d'une nouvelle tâche.

De la part des collaborateurs :

- Il est difficile de connaître l'historique du suivi d'un dossier lorsque l'on replace un collègue.
- La documentation n'est pas toujours bien rédigée. La date de dernière mise à jour n'est pas connue. Il existe plusieurs documents portant sur le même sujet.
- Il est difficile d'être au courant de toutes les actions planifiées par les autres collègues dans une journée. Il est devenu difficile de répondre efficacement aux projets sur la disponibilité d'un environnement de test.

La synthèse des résultats du sondage, chiffres et commentaires, établit une définition des besoins spécifique à chaque niveau hiérarchique dans l'équipe : le responsable de l'équipe et les collaborateurs.

Les besoins exprimés par le responsable de l'équipe sont :

- La traçabilité et les statistiques de l'activité de l'équipe.
- La quantification du travail des ressources.
- L'accès à tous les dossiers traités par l'équipe.
- La création et la gestion de dossier d'incident.
- L'accès à un résumé des informations et événements des environnements de test de manière quotidienne.
- La conversion et l'intégration de toute la documentation existante à GECKO.

Les besoins exprimés par les collaborateurs sont :

- Un outil optimal de suivi des incidents traités par l'équipe.
- La facilité d'accès à la documentation et la clarté des informations.
- Une disponibilité des informations et événements des environnements de test de manière quotidienne.
- Un moyen informatique de garder les notes de suivi d'incident.

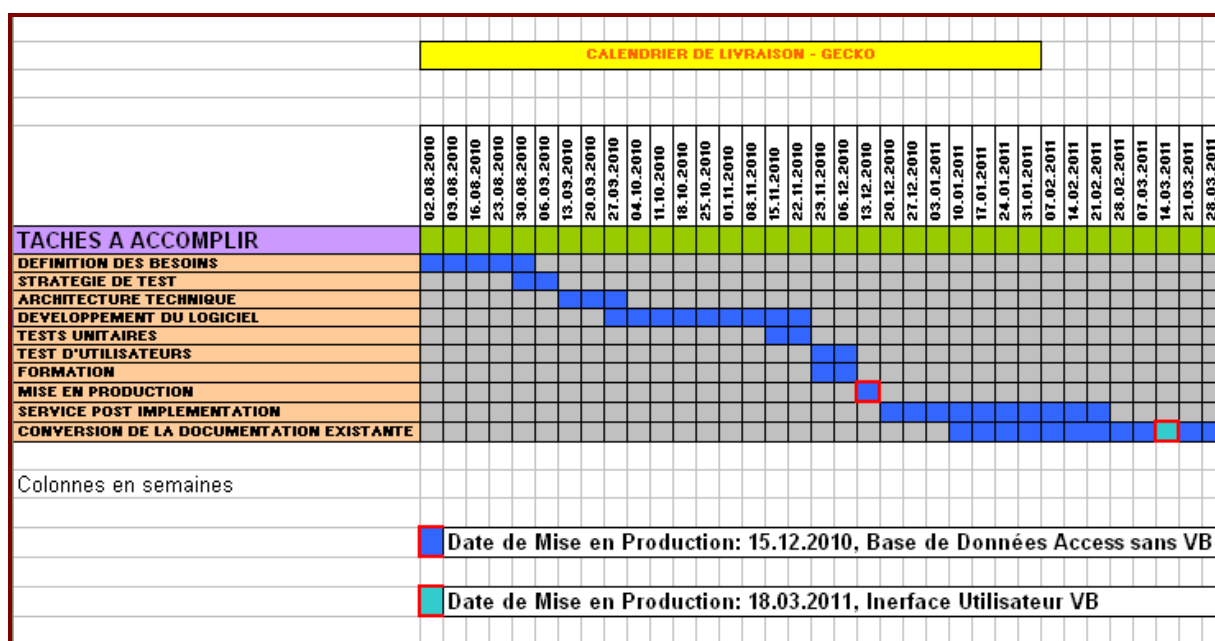
Les besoins en termes de capacité et de disponibilité du logiciel :

- Trois fonctions doivent être disponibles : La gestion des incidents, la gestion de la documentation, la disponibilité des événements quotidiens dans les environnements de test.
- En fonction des prévisions de croissance de l'équipe en termes de personnes, un maximum de dix connexions au logiciel est requis.
- Une interface d'utilisation conviviale et très peu chargée.
- Le temps nécessaire pour la saisie d'un dossier doit être minimal. Cinq minutes au maximum. La restitution des données doit être rapide lors de la consultation. Le temps de réponse du logiciel doit être optimal.
- En cas de panne de la base de données, le délai maximal d'indisponibilité de GECKO est de vingt-quatre heures.

La date requise de livraison du logiciel est le 15 Décembre 2010. Cette période est la moins chargée en activité pour l'équipe. Elle permettra de pouvoir se familiariser avec le logiciel et régler les problèmes de mise en production et de post-implémentations.

Une fois les étapes du projet définies, le calendrier suivant, représenté par un diagramme de Gantt a été établi et validé par le responsable de l'équipe.

Tableau 4.1.5 : Diagramme de Gantt du projet GECKO



2 - Les contraintes du projet GECKO

2.1 Les contraintes financières

Du fait de la nature du projet, interne à l'équipe Projects and Tests Environments Support, aucun budget n'est alloué. Le développement du logiciel Gecko se fera avec les outils informatiques dont la banque dispose déjà d'une licence.

2.2 Les contraintes techniques

2.2.1 – Les stations de connexion au logiciel

La connexion à GECKO se fera à partir des postes de travail actuels des membres de l'équipe. GECKO ne devra pas être en conflit avec les applications déjà installées sur les postes.

2.2.2– Les outils de développement de la base de données

Les outils utilisés pour le développement de GECKO seront Access 2003, SQL Server 2005 et Visual Basic 6.5.

2.2.3 – Le processus de qualité

Le développement et la mise en production de Gecko doivent suivre les standards de qualité de la banque et fournir les documents justificatifs de la stratégie de tests, des tests accomplis, tests unitaires et tests de validation des utilisateurs, de guide de déploiement et de stratégie de retour arrière en cas de problèmes d'utilisation du logiciel.

2.2.4 – La formation des utilisateurs

Une formation des utilisateurs est obligatoire avant la mise en production de GECKO, ainsi qu'une période d'adaptation. Enfin un justificatif devra être fourni prouvant l'adhérence de l'équipe à cet outil au plus tard le 15 Janvier 2011. Pour ce faire, l'administrateur de la base de données fera une extraction de données de la base et établira des statistiques d'utilisation par collaborateur.

2.3 Les contraintes de Ressources

Une seule ressource de l'équipe est allouée au projet GECKO à 25% soit 2 heures par jour. Cette ressource est chargée des tâches de toutes les étapes du projet. La progression du projet peut être affectée par des tâches plus prioritaires : les urgences, les imprévus et le support important devant être fourni à la Release Délivre Management du 05 Septembre 2010 au 22 Novembre 2010.

2.4 Les contraintes de temps

En prévision de la croissance de l'activité à partir du 02 Janvier 2010, la mise en production est fixée au 15 Décembre 2010. Une période de formation de deux semaines aura lieu avant la mise en production, plus deux semaines d'adaptation après la mise en production.

V - LES ARCHITECTURES FONCTIONNELLE ET TECHNIQUE DU LOGICIEL

La satisfaction des besoins exprimés par l'équipe nécessite une base de données et un système de gestion de base de données.

En fonction des contraintes qui sont imposées au projet, le logiciel GECKO sera développé à l'aide d'Access 2003. Afin d'augmenter les performances de la base de données Access, par exemple en cas de connexions simultanées de dix personnes, un moteur SQL Server 2005 sera couplé à GECKO.

L'interface utilisateur sera développée à l'aide de Visual Basic 6.5. Certains résultats d'exécution de requêtes dans GECKO seront présentés à l'aide Microsoft Word 2003.

La base de données sera accessible tous les jours de 7 heures du matin à 23 heures. Hors mis les périodes d'exécution des sauvegardes.

Une solution fonctionnelle a été définie à partir des informations données par les utilisateurs sur leurs attentes de l'outil GECKO. Lors de la conception du schéma fonctionnel, une attention particulière a été apportée à fournir un produit simple à utiliser, avec le minimum de champs possibles à remplir par l'utilisateur. Ce schéma a été accepté et validé par le sponsor du projet. La solution technique a été construite à la suite en se basant sur le schéma fonctionnel.

1 - Le schéma Fonctionnel de GECKO

Le schéma fonctionnel ci-après est composé de trois parties. C'est une représentation qui part d'une vue générale et qui rentre ensuite de plus en plus dans le détail de la base de données.

La première partie intitulée Workflow Fonctionnel de GECKO, présente le lien entre l'ordinateur de l'utilisateur et la base de données, puis la relation Client – Serveur entre la base de données ACCESS et le moteur SQL qui y sera rajouté.

La seconde partie intitulée Les fonctionnalités de la base de données GECKO, montre le fonctionnement d'un point de vue utilisateur, de la base de données. La première étape est l'identification de l'utilisateur. Ensuite selon son profil, la seconde étape montre les fonctions qui sont autorisées à la personne qui est connectée.

La troisième partie intitulée les tables de la base de données GECKO, présente les cinq tables qui contiendront les informations de la base données : Utilisateur, Activité, Connaissance, Evénement, Application.

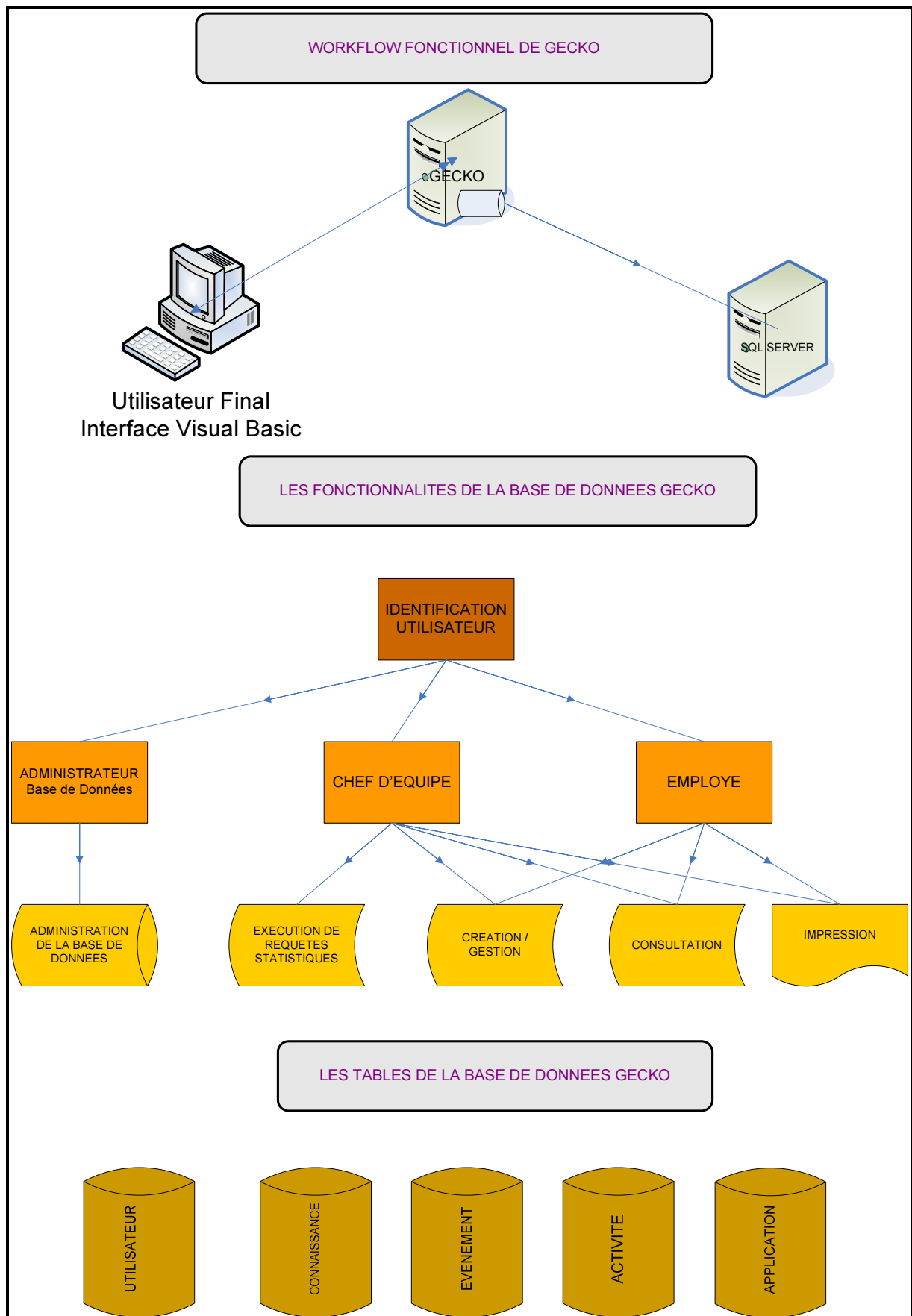


Figure V.I.I : Schéma fonctionnel de GECKO

1.1 L'identification de l'utilisateur

Trois profils type seront créés pour gérer les accès des utilisateurs de la base de données :

Le profil Administrateur :

L'administrateur de la base de données a accès à toutes les fonctionnalités disponibles aux autres utilisateurs. Il est le seul à pouvoir gérer la table UTILSATEUR où sont répertoriés les droits des utilisateurs. Il est également chargé de la maintenance de la base de données pour des tâches comme l'extension de taille, l'archivage et la sauvegarde de données, l'administration de compte utilisateur, la restitution de données en cas de perte, administration du moteur SQL Server...). Le profil Administrateur sera attribué au membre de l'équipe en charge de la maintenance de GECKO. Le processus de maintenance est décrit au chapitre Neuf du mémoire.

Le profil Chef d'équipe :

Ce profil est spécialement défini pour le responsable de l'équipe Projects and Tests Environments support. Ce n'est pas un profil administrateur. Cependant il permet d'exécuter des requêtes pour extraire des statistiques de la base de données.

Le profil Employé :

Ce profil est attribué par défaut à tous les membres de l'équipe. Il permet de :

- consulter les données en exécutant des requêtes par date, par numéro d'incident, ou par utilisateur.
- imprimer des dossiers ou des résultats de requête.
- créer, mettre à jour ou supprimer des données.

La figure ci – dessous montre un exemple de la mire de connexion proposée aux utilisateurs de GECKO. L'utilisateur devra rentrer son trigramme et son mot de passe pour accéder à l'application.

Trigramme	ABC
Mot de passe	xxxxxxxxxx

Figure V.I.II : Mire de connexion à GECKO

Le tableau ci – après montre un récapitulatif des droits d'accès à chaque table de la base de données.

Tableau 5.1.3 : Résumé des fonctionnalités accessibles selon le profil de connexion

FONCTIONNALITES	ADMINISTRATEUR	CHEF D'EQUIPE	EMPLOYE
TABLE UTILISATEUR			
CONSULTATION	oui	non	non
CREATION	oui	non	non
MODIFICATION	oui	non	non
SUPPRESSION	oui	non	non
IMPRESSION	oui	non	non
TABLE ACTIVITE			
STATISTIQUES	oui	oui	non
CONSULTATION	oui	oui	oui
CREATION	oui	oui	oui
MODIFICATION	oui	oui	oui
SUPPRESSION	oui	oui	oui
IMPRESSION	oui	oui	oui
CONSULTATION	oui	oui	oui
TABLE CONNAISSANCE			
CONSULTATION	oui	oui	oui
CREATION	oui	oui	oui
MODIFICATION	oui	oui	oui
SUPPRESSION	oui	oui	oui
IMPRESSION	oui	oui	oui
TABLE EVENEMENT			
CONSULTATION	oui	oui	oui
CREATION	oui	oui	oui
MODIFICATION	oui	oui	oui
SUPPRESSION	oui	oui	oui
IMPRESSION	oui	oui	oui
TABLE APPLICATION			
CONSULTATION	oui	oui	oui
CREATION	oui	oui	oui
MODIFICATION	oui	oui	oui
SUPPRESSION	oui	oui	oui
IMPRESSION	oui	oui	oui

1.2 Un menu d'accueil selon le profil connecté

Un menu d'accueil différent s'affichera suivant le profil qui se connecte à GECKO.

Le profil administrateur reçoit un menu d'accueil qui propose d'accéder à différentes fonctions :

- STATISTIQUES
- UTILISATEUR
- ACTIVITE
- DOCUMENTATION
- EVENEMENTS
- APPLICATION

La liste des évènements planifiés dans les environnements de test le jour de connexion s'affiche en permanence dans le menu d'accueil. Par exemple les environnements de test dont les données vont être rafraîchies depuis la production, les serveurs de tests sur lesquels sont effectuées des maintenances...

Le profil chef d'équipe reçoit un menu d'accueil qui propose d'accéder aux fonctions suivantes:

- STATISTIQUES
- ACTIVITE
- DOCUMENTATION
- EVENEMENTS
- APPLICATION

La liste des évènements planifiés dans les environnements de test le jour de connexion s'affiche en permanence dans le menu d'accueil du profil Chef d'équipe.

Le profil employé reçoit un menu d'accueil qui propose d'accéder à :

- ACTIVITE
- DOCUMENTATION
- EVENEMENTS
- APPLICATION

La liste des évènements planifiés dans les environnements de test le jour de connexion s'affiche en permanence dans le menu d'accueil du profil Employé.

1.3 Le descriptif de chacune des fonctionnalités proposées par le menu d'accueil

STATISTIQUES : Cette fonctionnalité destinée au profil Chef d'équipe, est un ensemble de requêtes destinées à extraire des données selon les critères de recherche suivants : date de début de période, date de fin de période, application, collaborateur, statut de dossier... Les critères de sélection peuvent être combinés afin d'obtenir des résultats plus spécifiques. L'impression est également disponible.

Par exemple, extraire le nombre de dossier traités par un employé sur une période déterminée et trier le résultat par application.

La figure ci-dessous présente un exemple de l'écran de la fonction STATISTIQUES.

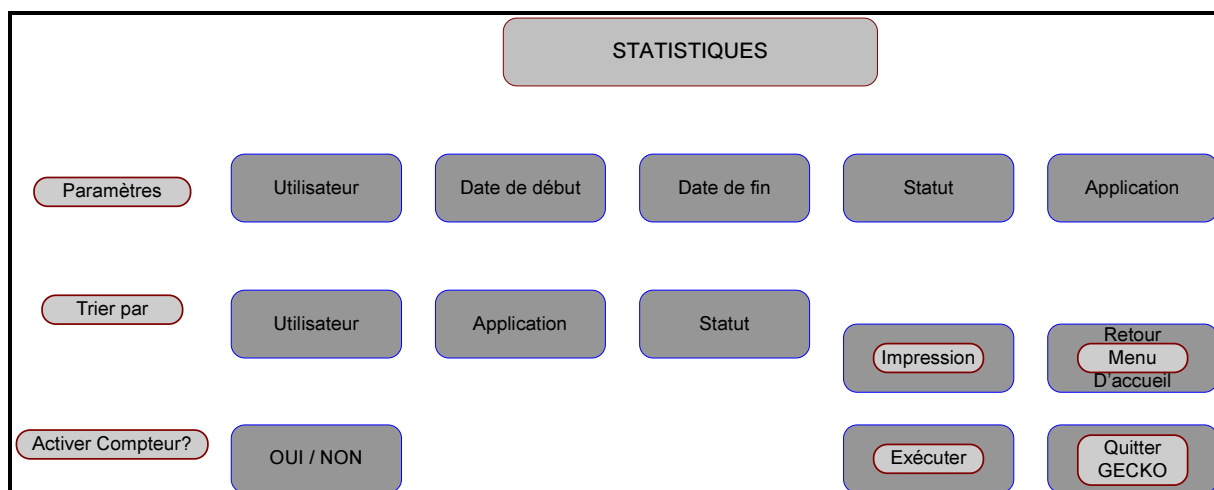


Figure V.I.IV : Menu de la fonction STATISTIQUES

ACTIVITE : La fonction ACTIVITE donne accès aux données relatives aux dossiers traités par l'équipe. Elle propose une création d'un nouveau dossier, la consultation de données selon des critères comme la date, la période, l'employé, l'application, le numéro de dossier.

La figure ci-dessous montre une représentation du menu de la fonction ACTIVITE.

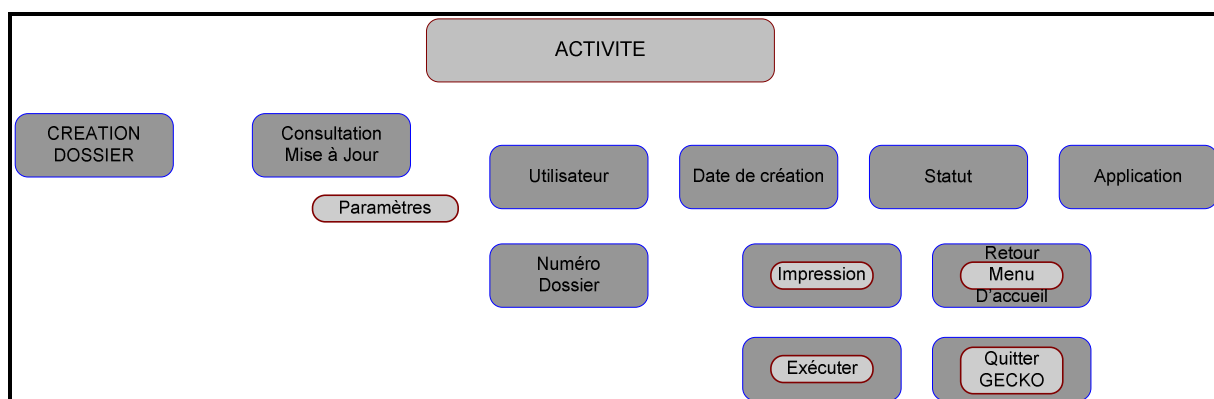


Figure V.I.V : Menu de la fonction ACTIVITE

DOCUMENTATION : La fonction DOCUMENTATION donne accès aux procédures, documents et base de connaissance de l'équipe. Elle propose la création d'une nouvelle documentation, la consultation de la base de connaissance selon les critères suivants : le nom de l'application ou l'utilisateur ayant créé le document. Les résultats seront présentés de manière chronologique selon la dernière date de modification.

La figure ci-dessous montre une représentation du menu de la fonction DOCUMENTATION.

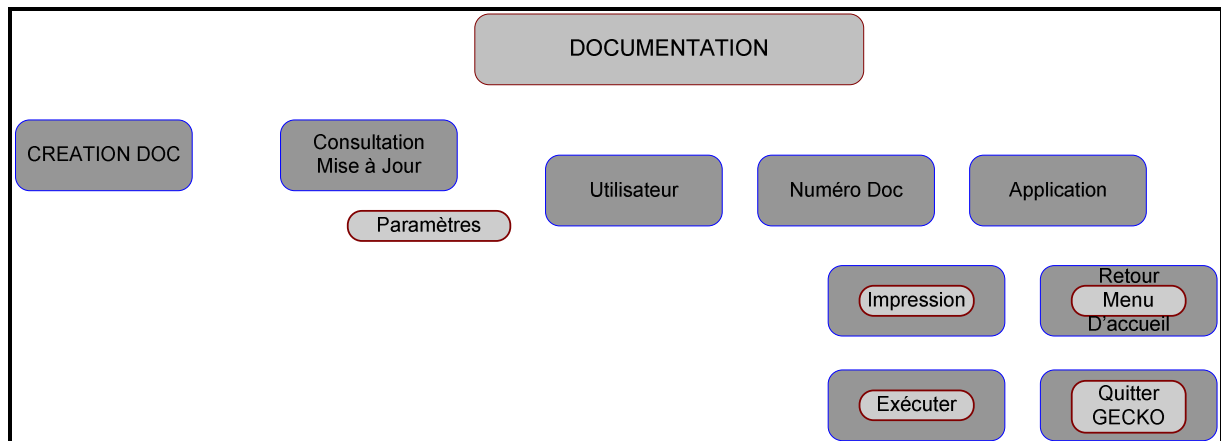


Figure V.I.VI : Menu de la fonction DOCUMENTATION

EVENEMENTS : La fonction EVENEMENTS donne accès à la base de données contenant les événements enregistrés par l'équipe. Elle propose la création d'un nouvel événement, la consultation de données selon des critères comme le numéro d'identification de l'évènement, la date à laquelle aura lieu l'évènement ou l'application concernée par l'évènement. Les résultats seront automatiquement triés par ordre chronologique de la date à laquelle aura lieu l'évènement.

La figure ci-dessous montre une représentation du menu de la fonction EVENEMENT.

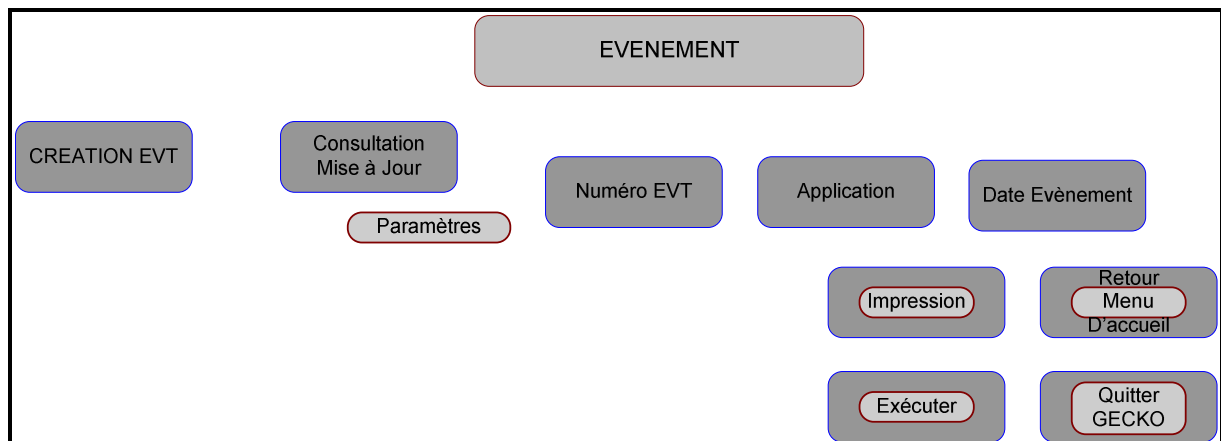


Figure V.I.VII : Menu de la fonction EVENEMENT

APPLICATION : La fonction APPLICATION donne accès à la base de données contenant les informations des applications existant dans les environnements de tests. Elle propose la création d'une nouvelle fiche d'informations sur une application, la consultation de données selon des critères comme le nom de l'application et / ou le trigramme de la personne qui a créé la fiche. Les résultats seront automatiquement triés par ordre alphabétique du nom de l'application.

La figure ci-dessous montre une représentation du menu de la fonction ACTIVITE.

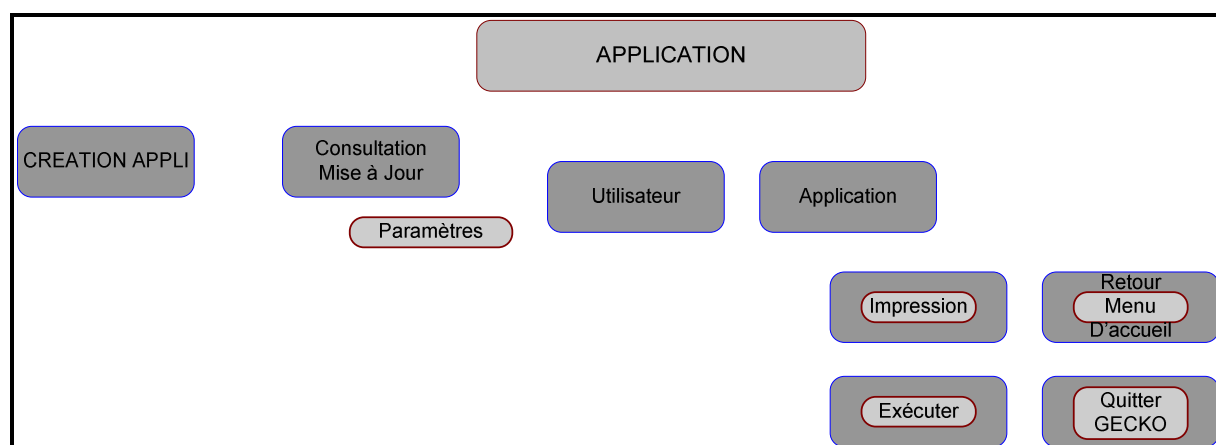


Figure V.I.VIII : Menu de la fonction APPLICATION

2 - L'architecture technique de GECKO

Les sources de la base de données GECKO seront hébergées sur un serveur Windows, dans un répertoire situé sur le disque dédié au département ITQTS.

2.1 Descriptions fonctionnelle et technique des tables constituant la base de données GECKO

La table UTILISATEUR

La table UTILISATEUR est accessible uniquement par l'administrateur de la base de données. Elle contient les identifiants (Nom, Prénom, Trigramme de connexion, Identifiant employé et type de profil) des personnes autorisées à accéder à GECKO.

Tableau 5.2.1 : Descriptif des champs de la table UTILISATEUR

Nom du Champ	Type de la donnée	Clé ?	Obligatoire ?	Descriptif de la donnée
TRIGRAMME	Texte sur 3 positions obligatoires	Oui - primaire	Oui	Le trigramme est un identifiant unique attribué à chaque employé de la banque
NOM	Texte sur maximum 25 positions	Non	Oui	Nom da personne qui crée le dossier
PRENOM	Texte sur maximum 20 positions	Non	Oui	Prénom de la personne qui crée le dossier
ID EMPLOYE	Texte sur 8 positions obligatoires	Oui	Oui	Identifiant unique attribué à chaque employé de la banque
TYPE PROFIL	Choix dans un menu déroulant	Non	Oui	Type de profil de l'utilisateur : Employé, chef d'équipe ou administrateur

Exemple d'une occurrence de la table UTILISATEUR

TRIGRAMME	NOM	PRENOM	ID EMPLOYE	TYPE PROFIL
WLU	LAWSON-BODY	ELINA	11156548	EMPLOYEE

Elina LAWSON-BODY*Mémoire d'Ingénieur*CNAM DE SAINT GENIS POUILLY*2010-2011

GECKO

La table ACTIVITE

La table ACTIVITE est accessible par tous les utilisateurs de GECKO. Elle contient les informations relatives aux dossiers traités et saisis par l'équipe.

Tableau 5.2.2 Descriptif des champs de la table ACTIVITE

Nom du Champ	Type de la donnée	Clé ?	Obligatoire ?	Descriptif de la donnée
NUMERO DOSSIER	Entier – Numéro Automatique	Oui - Primaire	Oui	Identifiant unique de l'occurrence, automatiquement mis par l'application
UTILISATEUR	Texte sur 3 positions obligatoires – Menu déroulant	non	Oui	Identifiant de la personne qui crée le dossier. C'est un identifiant unique attribué à chaque employé de la banque
DATE CREATION	Date / Heure	non	Oui	Date de la création du dossier
HEURE CREATION	Date / Heure	non	Oui	Heure de la création du dossier
STATUT	Texte – Menu déroulant	non	Oui	Statut du dossier, ouvert, fermé ou suspendu
CLIENT	CHAR sur 3 positions obligatoires	non	Oui	Identifiant de la personne à l'origine de la demande. C'est un identifiant unique attribué à chaque employé de la banque
APPLICATION	Texte – Menu déroulant	non	Oui	Nom de l'application concernée par la demande. Une liste extraite de la table APPLICATION est proposée.
TITRE	Texte	non	Oui	Court descriptif de la demande
DESCRIPTIF SUIVI ET RESOLUTION	Texte	non	Oui	Historique et descriptif du suivi et de la résolution de la demande

Exemple d'une occurrence de la table ACTIVITE

NUMERO DOSSIER	00345
UTILISATEUR	WLU
DATE CREATION	22.10.2010
HEURE CREATION	09 :00
STATUS	OPEN
CLIENT	EJB
APPLICATION	R-ISLAND
DESCRIPTIF REQUETE	PASSWORD RESET ON DATABASE
DESCRIPTIF SUIVI ET RESOLUTION	REQUEST SENT TO DBA

La table CONNAISSANCE

La table CONNAISSANCE est accessible par tous les utilisateurs de GECKO. Elle contient les informations relatives à la documentation et aux procédures de l'équipe.

Tableau 5.2.3 : Descriptif des champs de la table CONNAISSANCE

Nom du Champ	Type de la donnée	Clé ?	Obligatoire	Descriptif de la donnée
NUMERO DOC	Entier – Numéro Automatique	Oui - Primaire	Oui	Identifiant unique de l'occurrence, automatiquement mis par l'application
APPLICATION	Texte – Menu déroulant	non	Oui	Nom de l'application concernée par la demande. Une liste extraite de la table APPLICATION est proposée.
UTILISATEUR	Texte sur 3 positions obligatoires – Menu déroulant	non	Oui	Identifiant de la personne qui crée le document
DATE CREATION	Date / Heure	non	Oui	Date de la création du dossier
TITRE	Texte max 50 caractères	non	Oui	Titre de la documentation
RESUME	Texte	non	Non	Résumé de la documentation
DESCRIPTIF LONG	Texte	non	Oui	Documentation

Exemple d'une occurrence de la table CONNAISSANCE

NUMERO DOC	000012
APPLICATION	EQUALIZER
UTILISATEUR	WLU
DATE CREATION	22.10.2010
TITRE	PMS BATCH
RESUME	Préparation, Exécution et Suivi de Batch PMS
DESCRIPTIF LONG	...

La table APPLICATION

La table APPLICATION est accessible par tous les utilisateurs de GECKO. Elle contient les informations relatives aux applications présentes dans les environnements de tests, telles que nom de serveurs, nom de base de données...

Tableau 5.2.4 : Descriptif des champs de la table APPLICATION

Nom du Champ	Type de la donnée	Clé ?	Obligatoire ?	Descriptif de la donnée
NUMERO APPLI	Entier – Numéro Automatique	Oui - Primaire	Oui	Identifiant unique de l'occurrence, automatiquement mis par l'application
NOM APPLICATION	Texte max 30 caractères	non	Oui	Nom de l'application
UTILISATEUR	Texte sur 3 positions obligatoires – Menu déroulant	non	Oui	Identifiant de la personne qui saisit les informations relatives à l'application
DETAILS	Texte	non	Oui	Informations sur les serveurs, base de données, interfaçages relatifs à l'application

Exemple d'une occurrence de la table APPLICATION

NUMERO APPLI	0032
NOM APPLICATION	BIMAS
UTILISATEUR	WLU
DETAILS	DEV : nom serveur ibiza ; DATACOM ; Interface MQ Series

La table EVENEMENT

La table EVENEMENT est accessible par tous les utilisateurs de GECKO. Elle contient les informations relatives aux événements planifiés dans les environnements de test. L'objectif étant d'extraire quotidiennement ces informations et de les mettre à disposition de l'équipe pour une durée de 24 heures.

Tableau 5.2.5 : Descriptif des champs de la table EVENEMENT

Nom du Champ	Type de la donnée	Clé ?	Obligatoire ?	Descriptif de la donnée
NUMERO EVENEMENT	Entier – Numéro Automatique	Oui - Primaire	Oui	Identifiant unique de l'occurrence, automatiquement mis par l'application
APPLICATION	Texte – Menu déroulant	non	Oui	Nom de l'application concernée par la demande. Une liste extraite de la table APPLICATION est proposée.
UTILISATEUR	Texte sur 3 positions obligatoires – Menu déroulant	non	Oui	Identifiant de la personne qui créé le document
DATE CREATION	Date / Heure	non	Oui	Date de la création du dossier
HEURE CREATION	Date / Heure	non	Oui	Heure de la création du dossier
DATE EVENEMENT	Date / Heure	non	Oui	Date à laquelle aura lieu l'évènement dans l'environnement
TITRE	Texte max 20 caractères	non	Oui	Titre de l'évènement
DETAILS	Texte	non	Non	Détails de l'évènement

Exemple d'une occurrence de la table EVENEMENT

NUMERO EVENT	003
APPLICATION	BIMAS
UTILISATEUR	WLU
DATE CREATION	22.10.10
HEURE CREATION	11 :03
TITRE	REFRESH UGVA1B
DETAILS	Projet Client : SWIFT, FCG, backup post batch 30.09.2009 avec DB100

2.2 Le modèle Conceptuel de Données

Le modèle conceptuel de données est une représentation précise de chaque élément constituant une base de données (données, tables et relations entre tables). En d'autres termes c'est une description de la structure des données que comporte une base.

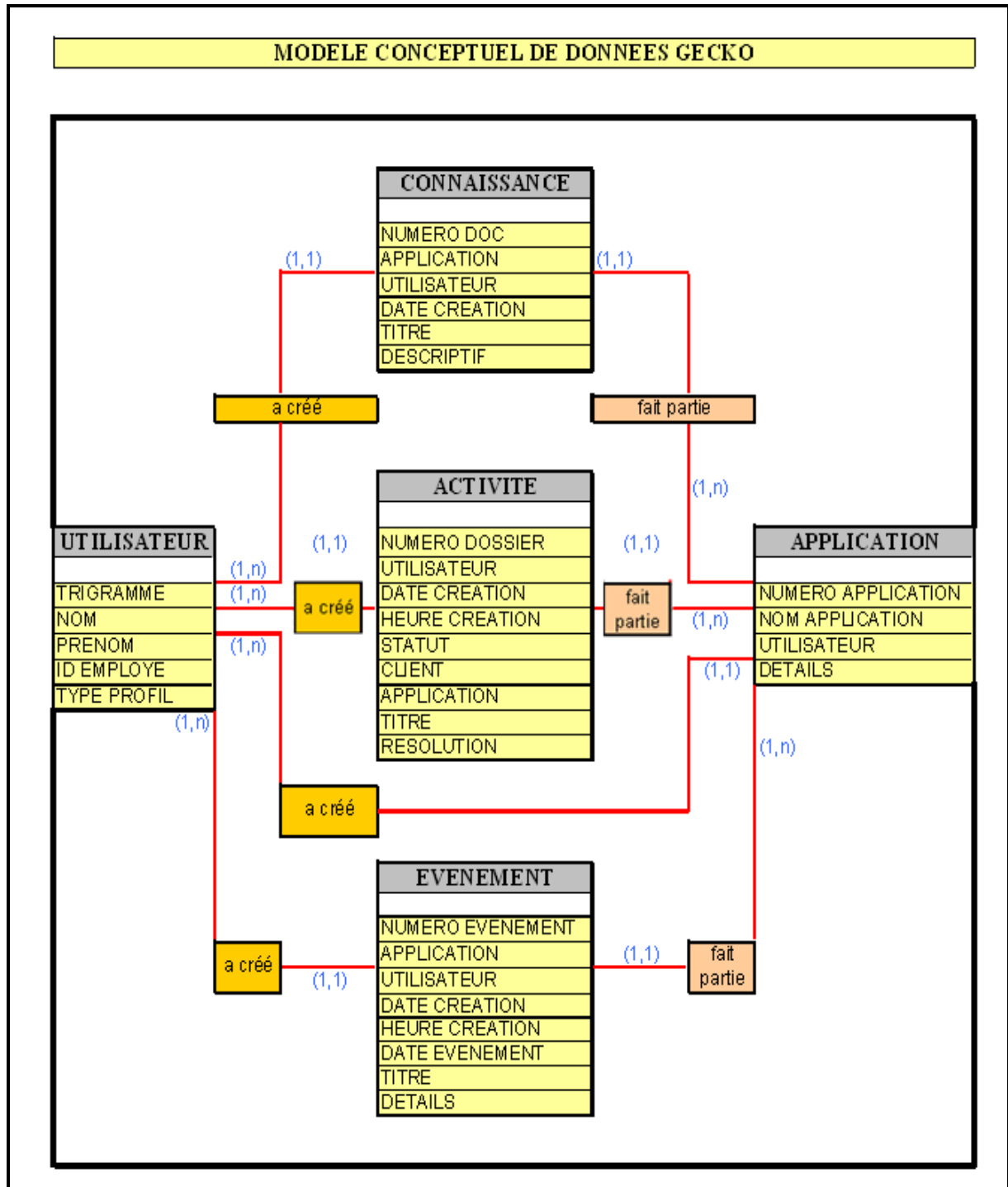
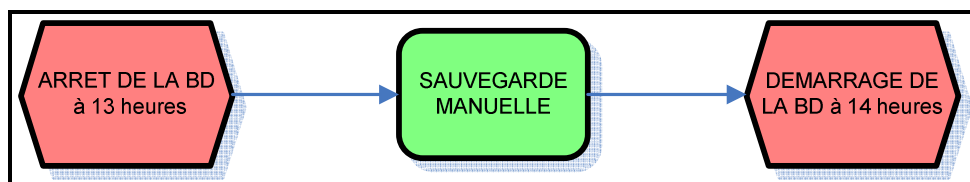


Figure V.II.VI : Modèle conceptuel de la base de données GECKO

2.3 La méthode de sauvegarde de GECKO

Un processus de sauvegarde de la base de données GECKO sera mis en place. Manuelles dans un premier temps et planifiées à 13 heures quotidiennement, du lundi au vendredi, les sauvegardes seront par la suite automatisées et se déclencheront tous les jours à 23 heures également du lundi au vendredi. Les sauvegardes manuelles seront effectuées par la personne en charge de la maintenance de GECKO dans l'équipe Projects and Tests Environments Support. Les sauvegardes automatiques seront confiées à l'équipe d'administrateurs de bases de données de la banque à partir du 2 septembre 2011.

Le processus de Sauvegarde manuelle



Le processus de Sauvegarde automatique

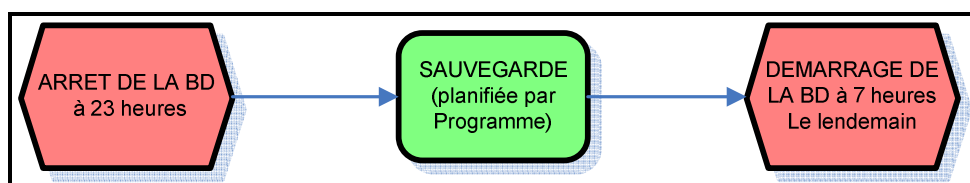


Figure V.II.VII : Processus de sauvegarde GECKO : manuel et automatique

La durée de garde des sauvegardes

Les sauvegardes de la base de données seront hébergées sur un disque différent de celui où se trouve la base de données. Ses sauvegardes seront accessibles uniquement par l'administrateur de la base de données.

Tableau 5.2.8 Durées de garde des copies de la base de données GECKO

TYPES DE SAUEGARDES	DUREE DE GARDE
Journalières – différentielles	10 jours
Hebdomadaires	30 jours
Mensuelles	60 jours
Trimestrielles	180 jours
Annuelles	3 ans

2.4 L'hébergement de la Base de Données

L'espace disque nécessaire pour héberger la base de données est évalué à 1GB pour la première année de production. Les années suivantes une augmentation de 300 MB sera nécessaire.

Ces chiffres ont été obtenus grâce à une étude de l'évolution de la volumétrie de la documentation et du total des emails traités par année, sur une période de cinq ans.

2.5 La conversion de la documentation existante

La documentation et les procédures existantes seront converties et intégrées à la table CONNAISSANCE. Les procédures et diverses documentations sont estimées à 8000 fichiers de format Word et Excel incluant des schémas Visio, et associées à 89 dossiers correspondant chacun à une application.

Il est planifié de convertir 4 applications par semaine, soit une application par collaborateur. Le processus consiste à passer en revue de tous les documents contenus dans le dossier de l'application, éliminer les informations redondantes et créer deux fichiers consolidés de format Word :

- Un fichier *nom_application_APPLICATION* contenant les informations techniques relatives à l'application. Les données de ce fichier seront saisies dans la table APPLICATION. Par exemple pour l'application BIMAS, le fichier sera nommé BIMAS_APPLICATION
- Un fichier *nom_application_CONNAISSANCE* contenant la connaissance et la documentation de l'application, dont les données seront saisies dans la table CONNAISSANCE. Par exemple pour l'application BIMAS, le fichier sera nommé BIMAS_CONNAISSANCE

Ensuite le contenu de chaque document est copié et collé dans le formulaire de la table correspondante.

Le contenu du document APPLICATION est copié et collé dans le champ DETAILS d'un nouvel enregistrement de la table APPLICATION. Le nom de l'application est saisi dans le champ NOM APPLICATION.

Le contenu du document CONNAISSANCE est copié et collé dans le champ DESCRIPTIF d'un nouvel enregistrement de la table CONNAISSANCE. Le nom de la documentation est saisi dans le champ TITRE.

2.6 La gestion des accès utilisateur et des mots de passe

La définition et la gestion des droits utilisateurs et des mots de passe seront sous la responsabilité de l'administrateur de la base de données GECKO. Par défaut les profils attribués seront EMPLOYE pour tous les utilisateurs sauf pour le responsable de l'équipe. Ce dernier aura un profil CHEF D'EQUIPE.

2.7 Le couplage de la base de données ACCESS au moteur SQL Server

La base de données GECKO sera uniquement gérée par ACCESS 2003, les douze premiers mois suivant la mise en production. Cette période est nécessaire pour la mise en route du logiciel, la conversion des documents existants et les corrections des fonctionnalités.

SQL Server étant requis pour augmenter les performances de la base, on estime que ce besoin va se manifester à la fin de l'année 2011. L'équipe comprendra 8 personnes, toute la documentation sera dans GECKO et l'utilisation du logiciel sera devenue systématique. Le nombre de requêtes aura considérablement augmenté.

Le couplage de la base de données ACCESS à SQL Server se fera en commun avec l'équipe d'administration des bases de données de la banque. Les informations nécessaires sont la structure technique de la base de données (nom et caractéristiques des tables, des champs,...) l'estimation de l'espace disque nécessaire à l'historique, la gestion des accès des utilisateurs.

SQL Server est un système de gestion de base de données relationnelles, puissant et fiable, fourni par Microsoft. Il offre des avantages non négligeables lorsqu'il est connecté à une base de données Access. En effet son moteur permet de gérer un volume de données plus important qu'Access, surtout en cas d'un nombre de connexions simultanées s'approchant de dix. SQL Server possède des fonctionnalités permettant d'augmenter les performances de la base de données, et de gérer la sécurité des données. De plus il est facile à déployer, ce qui limite les ressources nécessaires à l'installation et à la maintenance.

2.8 La sécurité des données

Les données contenues dans les environnements de tests sont cryptées. Les informations sensibles contenues dans la base de données GECKO sont celles de la table APPLICATION. En effet, les détails de l'application sont le nom et les informations des serveurs et bases de données hébergeant l'application. Bien que ce soient tous des éléments de tests, il est indispensable de sécuriser la base de données selon les normes qui sont définies par la banque.

L'accès à la base de données étant possible de trois manières, les points définis ci-dessous expliquent la manière dont chaque accès est sécurisé :

- Via les postes des utilisateurs : La sécurité est appliquée avec la création de session utilisateur individuelle, protégée par un mot de passe.
- Via le serveur Windows hébergeant GECKO : La partition sur laquelle est stocké GECKO est accessible par le département ITQTS, soit 14 personnes au total. Ne possédant pas de compte sur GECKO, aucune personne étrangère à l'équipe ne peut accéder à la base de données même en faisant partie du même département. La seule option serait de pirater l'accès à la base.
- Via SQL server : Le serveur hébergeant le moteur SQL Server 2005 est géré par l'équipe d'administrateurs de bases de données de la banque. Plusieurs applications couplées à SQL Server 2005 y sont installées. L'accès est exclusivement autorisé à cette équipe.

VI - DEVELOPPEMENT DE LA SOLUTION RETENUE

Le développement de l'application GECKO suivra le schéma technique défini au chapitre précédent. La performance de la base de données reste l'objectif premier dans la définition de tous les éléments.

La création des éléments débutera par les tables, suivront les requêtes, les formulaires, les états, les macros, les graphiques, les menus de l'interface utilisateur et la définition des accès à la base selon les profils des utilisateurs.

Une convention de nommage est respectée, afin d'éviter toute confusion au niveau des noms des éléments de la base de données.

La nomenclature des tables :

Le nom des tables est défini en lettres majuscules et sans accent. Chaque champ des tables est écrit en lettres minuscules, sans accents, sans tirets et sans espaces. Par exemple le champ contenant le numéro de dossier dans la table ACTIVITE, est nommé numerodossier.

La nomenclature des requêtes :

La lettre 'q' ('q' comme 'query', en anglais requête) préfixe le nom des requêtes, qui est écrit en lettres minuscules, sans accents, sans tirets et sans espaces. Par exemple la requête qui extrait des données de la table activité est nommée 'qactivite'.

Lors de la rédaction des requêtes, les champs sont identifiés par le nom de la table dont ils font partie, puis par le nom du champ. Les deux mots sont séparés par un point. Par exemple dans la requête 'qactivite', la première instruction est ainsi rédigée :

SELECT ACTIVITE.numerodossier, ACTIVITE.statut, ACTIVITE.client, ACTIVITE.titre, ACTIVITE.application, ACTIVITE.datecreation, ACTIVITE.resolution

La nomenclature des formulaires :

Le nom de chaque formulaire est rédigé en minuscules, sans espaces, sans accents et préfixé de la lettre 'f'. Par exemple le formulaire pour saisir les données de la table APPLICATION est nommé 'fapplication'.

La nomenclature des états:

Le nom de chaque état est rédigé en minuscules, sans espaces, sans accents et préfixé de la lettre 'r' ('r' comme 'report', en anglais état). Par exemple l'état utilisé pour afficher le résultat de la requête 'qactivite' se nomme 'ractivite'.

La nomenclature des macros:

Le nom de chaque macro est rédigé en minuscules, sans espaces, sans accents et préfixé de la lettre 'm'. Par exemple la macro exécutée dans le formulaire 'fapplication' pour revenir au menu d'accueil est nommée 'mbacktomenu'.

La nomenclature des graphiques:

Le nom de chaque graphique est rédigé en minuscules, sans espaces, sans accents et préfixé de la lettre 'g'. Par exemple le graphique obtenu à partir des statistiques de l'équipe est nommée 'gstategie'.

1 - La définition des tables

Les tables seront composées d'un nombre minimal de champs afin de respecter deux directives : Un temps de saisie court pour l'équipe et un temps optimal de restitution des données.

1.1 La table ACTIVITE

C'est la table la plus importante de la base, car elle contient les données qui seront plus sollicitées que celles des autres tables.

Elle contient neuf champs dont trois automatiques et trois en menu déroulant :



Field Name	Data Type	Description
numerodossier	AutoNumber	Numéro de dossier
utilisateur	Text	Trigramme de l'utilisateur
datecreation	Date/Time	Date de la création du dossier
heurecreation	Date/Time	Heure de la création du dossier
statut	Text	Statut du dossier
client	Text	Trigramme de la personne à l'origine de la demande
application	Text	Application concernée par la demande
titre	Text	Titre du dossier
resolution	Memo	Descriptif du suivi et de la résolution

Figure VI.1.1 : Définition de la table ACTIVITE

Le champ numerodossier : Ce champ obligatoire, de type entier, contient le numéro de chaque dossier créé dans la table activité. Afin de garantir l'unicité de chaque dossier, le numéro est attribué de manière automatique par Access. Ce champ est également défini comme clé primaire

Le champ utilisateur : Ce champ obligatoire, de type texte et autorisé sur 3 champs, est alimenté par la table utilisateur afin de s'assurer que la personne qui crée le dossier fait partie de l'équipe. A ce niveau un contrôle n'est pas effectué entre le trigramme de l'utilisateur et l'identifiant qui s'est connecté à la base. Cette sécurité n'est pas jugée utile car toute l'équipe accède à cette table avec les mêmes droits, et les collègues gardent la possibilité de créer un dossier en utilisant une connexion qui n'est pas la leur. En cas d'oubli de mot de passe par exemple.

Les champs datecreation et heurecreation : Ces champs obligatoires, de type date et heure sont remplis de manière automatique à la création d'un dossier. Cette précaution est destinée à éviter une erreur de saisie dans la date et l'heure qui fausserait les statistiques et la consultation. C'est également un gain de temps pour l'utilisateur.

Le champ statut : Champ obligatoire, de type texte, il est composé d'un menu déroulant qui propose un choix à l'utilisateur : OUVERT, FERME, SUSPENDU.

Le champ client : Champ obligatoire, de type texte, limité sur 3 caractères et en mode majuscules.

Le champ application : Ce champ obligatoire, de type texte, est lié à la table APPLICATION et présente un menu déroulant de choix à l'utilisateur.

Les tables ACTIVITE et APPLICATION n'étant pas lié par des clés, une requête a été créée pour afficher dans le menu déroulant de ce champ, la liste des noms d'applications extraite de la table APPLICATION. Cette règle oblige l'utilisateur à renseigner la table APPLICATION au cas où le nom de l'application n'existe pas encore.

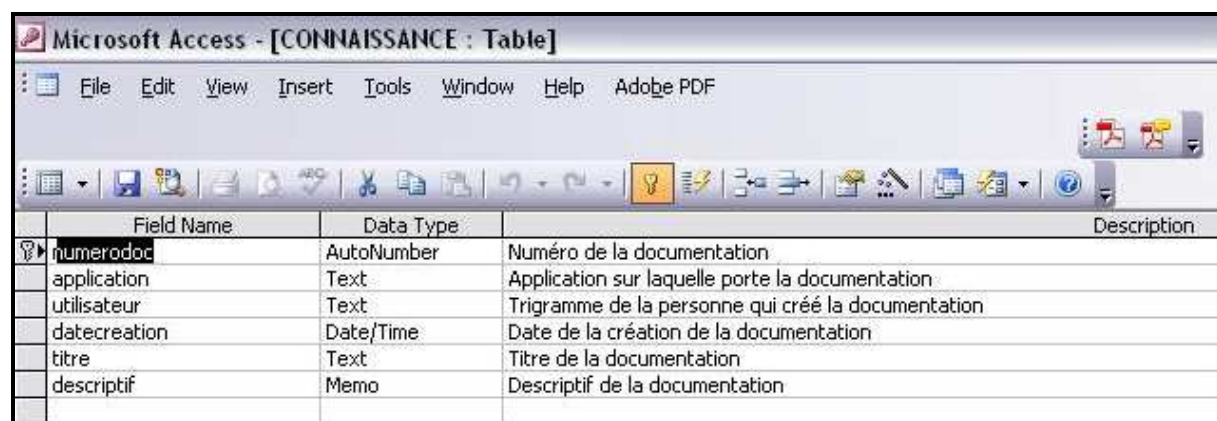
**Select APPLICATION.nomapplication
From APPLICATION**

Le champ titre : Ce champ de type texte est obligatoire et est limité à 50 caractères en mode majuscules.

Le champ résolution : Ce champ de type memo, n'est pas obligatoire. En effet les descriptif de résolution de certains problèmes est évident à tous les membres de l'équipe. Le type memo permet la saisie de texte long contrairement au type texte qui demande une définition de longueur de champ.

1.2 La table CONNAISSANCE

Elle contient six champs dont un automatique et deux en menu déroulant :



Field Name	Data Type	Description
numerodoc	AutoNumber	Numéro de la documentation
application	Text	Application sur laquelle porte la documentation
utilisateur	Text	Trigramme de la personne qui crée la documentation
datecreation	Date/Time	Date de la création de la documentation
titre	Text	Titre de la documentation
descriptif	Memo	Descriptif de la documentation

Figure VI.I.II : Définition de la table CONNAISSANCE

Le champ numerodoc : Ce champ obligatoire, de type entier est automatiquement assigné par Access. C'est une clé primaire de la table.

Le champ application : Champ obligatoire, de type texte, est lié à la table application et présente un menu déroulant de choix à l'utilisateur.

Le champ utilisateur : Champ obligatoire, de type texte sur 3 caractères, il est lié à la table utilisateur. La règle de gestion est la même que celle de la table ACTIVITE.

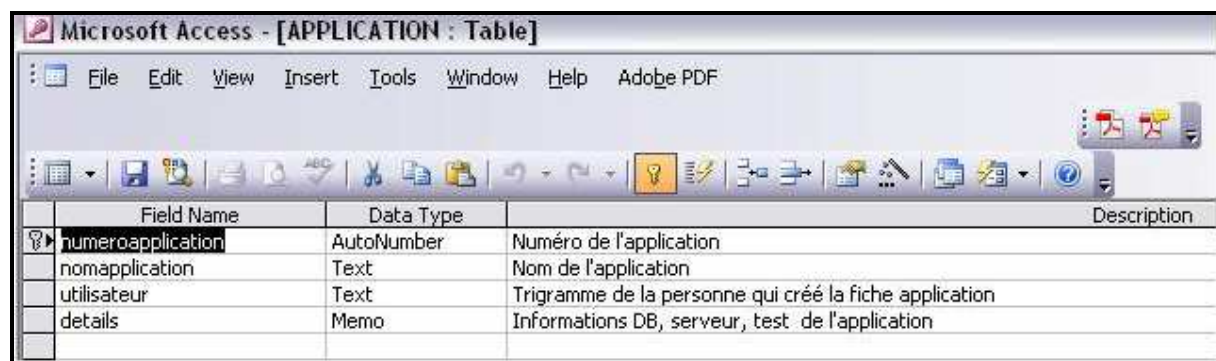
Le champ datecreation : Champ obligatoire, de type date, automatiquement rempli PAR ACCESS à la création d'une nouvelle occurrence.

Le champ titre : Champ obligatoire, de type texte et de format majuscule.

Le champ descriptif : Champ obligatoire de type memo.

1.3 La table APPLICATION

La table APPLICATION contient quatre champs dont un automatique et un menu déroulant.



The screenshot shows the Microsoft Access interface with the table 'APPLICATION' defined. The table has four fields: 'numeroapplication' (AutoNumber), 'nomapplication' (Text), 'utilisateur' (Text), and 'details' (Memo). The descriptions are: 'Numéro de l'application', 'Nom de l'application', 'Trigramme de la personne qui crée la fiche application', and 'Informations DB, serveur, test de l'application'.

Field Name	Data Type	Description
numeroapplication	AutoNumber	Numéro de l'application
nomapplication	Text	Nom de l'application
utilisateur	Text	Trigramme de la personne qui crée la fiche application
details	Memo	Informations DB, serveur, test de l'application

Figure VI.I.III : Définition de la table APPLICATION

Le champ numeroapplication : Champ obligatoire, de type entier et rempli de manière automatique par ACCESS à la création d'un nouvel enregistrement. Ce champ constitue la clé primaire de la table APPLICATION.

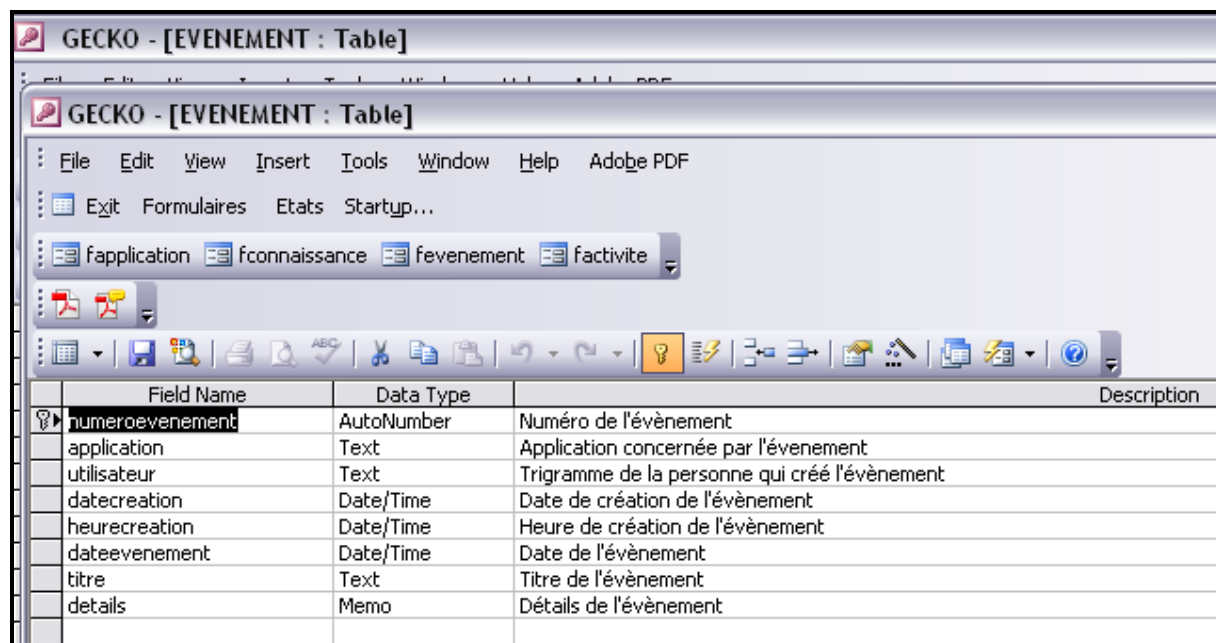
Le champ nomapplication : Champ obligatoire de type texte sur 50 caractères et de format majuscule.

Le champ utilisateur : Champ obligatoire de type texte sur trois caractères et alimenté par la table UTILISATEUR.

Le champ details : Champ obligatoire de type memo.

1.4 La table EVENEMENT

La table EVENEMENT est constituée de sept champs dont trois automatiques et deux alimentés par deux autres tables.



The screenshot shows the Microsoft Access interface with the table 'EVENEMENT' defined. The table has seven fields: 'numeroevenement' (AutoNumber), 'application' (Text), 'utilisateur' (Text), 'datecreation' (Date/Time), 'heurecreation' (Date/Time), 'dateevenement' (Date/Time), and 'titre' (Text). The descriptions are: 'Numéro de l'évènement', 'Application concernée par l'évènement', 'Trigramme de la personne qui crée l'évènement', 'Date de création de l'évènement', 'Heure de création de l'évènement', 'Date de l'évènement', and 'Titre de l'évènement'.

Field Name	Data Type	Description
numeroevenement	AutoNumber	Numéro de l'évènement
application	Text	Application concernée par l'évènement
utilisateur	Text	Trigramme de la personne qui crée l'évènement
datecreation	Date/Time	Date de création de l'évènement
heurecreation	Date/Time	Heure de création de l'évènement
dateevenement	Date/Time	Date de l'évènement
titre	Text	Titre de l'évènement
details	Memo	Détails de l'évènement

Figure VI.I.IV : Définition de la table EVENEMENT

Le champ numeroevenement : Champ obligatoire, de type entier et rempli de manière automatique par ACCESS à la création d'un nouvel enregistrement. Ce champ constitue la clé primaire de la table EVENEMENT.

Le champ application : Champ obligatoire de type texte sur 50 caractères et de format majuscule. Ce champ est un menu déroulant alimenté par la table APPLICATION.

Le champ utilisateur : Champ obligatoire de type texte sur trois caractères et alimenté par la table UTILISATEUR.

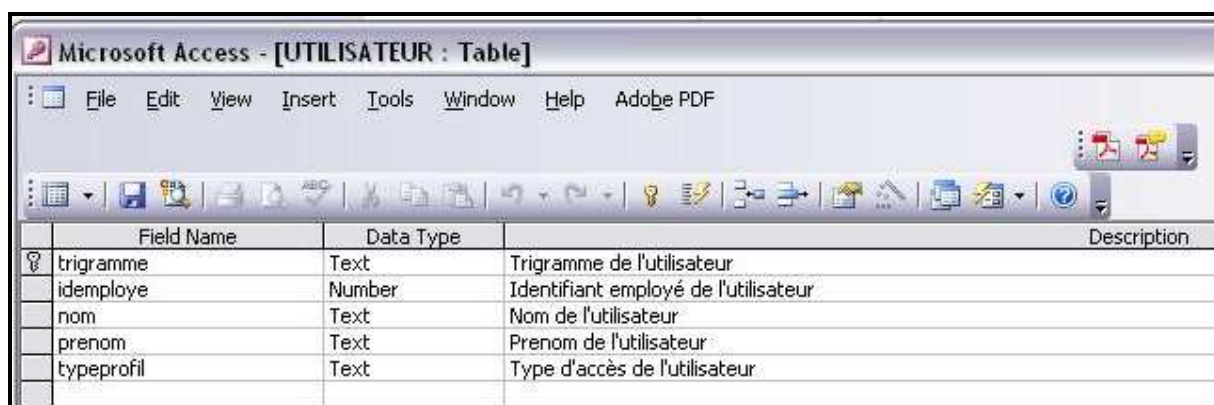
Les champs datecreation et heurecreation : Ces champs obligatoires, de type date et heure sont remplis de manière automatique à la création d'un nouvel enregistrement. Cette précaution est destinée à éviter une erreur de saisie dans la date et l'heure qui fausserait les statistiques et la consultation. C'est également un gain de temps pour l'utilisateur.

Le champ titre : Champ obligatoire de type texte sur cinquante caractères et en format majuscule.

Le champ details : Champ non obligatoire de type memo

1.5 La table UTILISATEUR

La table UTILISATEUR contient un champ au menu déroulant et aucun champ automatique et est accessible uniquement par l'administrateur de la base de données.



The screenshot shows the Microsoft Access interface for the 'UTILISATEUR' table. The table structure is as follows:

Field Name	Data Type	Description
trigramme	Text	Trigramme de l'utilisateur
idemploye	Number	Identifiant employé de l'utilisateur
nom	Text	Nom de l'utilisateur
prenom	Text	Prenom de l'utilisateur
typeprofil	Text	Type d'accès de l'utilisateur

Figure VI.I.V : Définition de la table UTILISATEUR

Le champ trigramme : Champ obligatoire de type texte, sur trois caractères et en format majuscules.

Le champ idemploye : Champ obligatoire, de type entier

Le champ nom : Champ obligatoire de type texte sur 20 caractère en majuscules

Le champ prenom : Champ obligatoire de type texte sur 20 caractère en majuscules

Le champ typeprofil : Champ obligatoire de type texte, proposant un menu déroulant.

1.6 Les relations entre les tables

Une fois les règles de validation des champs et les clés primaires définies dans les tables, la définition des relations entre les tables se fait à l'aide des clés primaires. L'intégrité référentielle est utilisée pour éviter que des données non cohérentes soient enregistrées dans les tables. Comme par exemple, créer un dossier activité avec un identifiant d'un utilisateur qui ne fait pas partie de l'équipe.

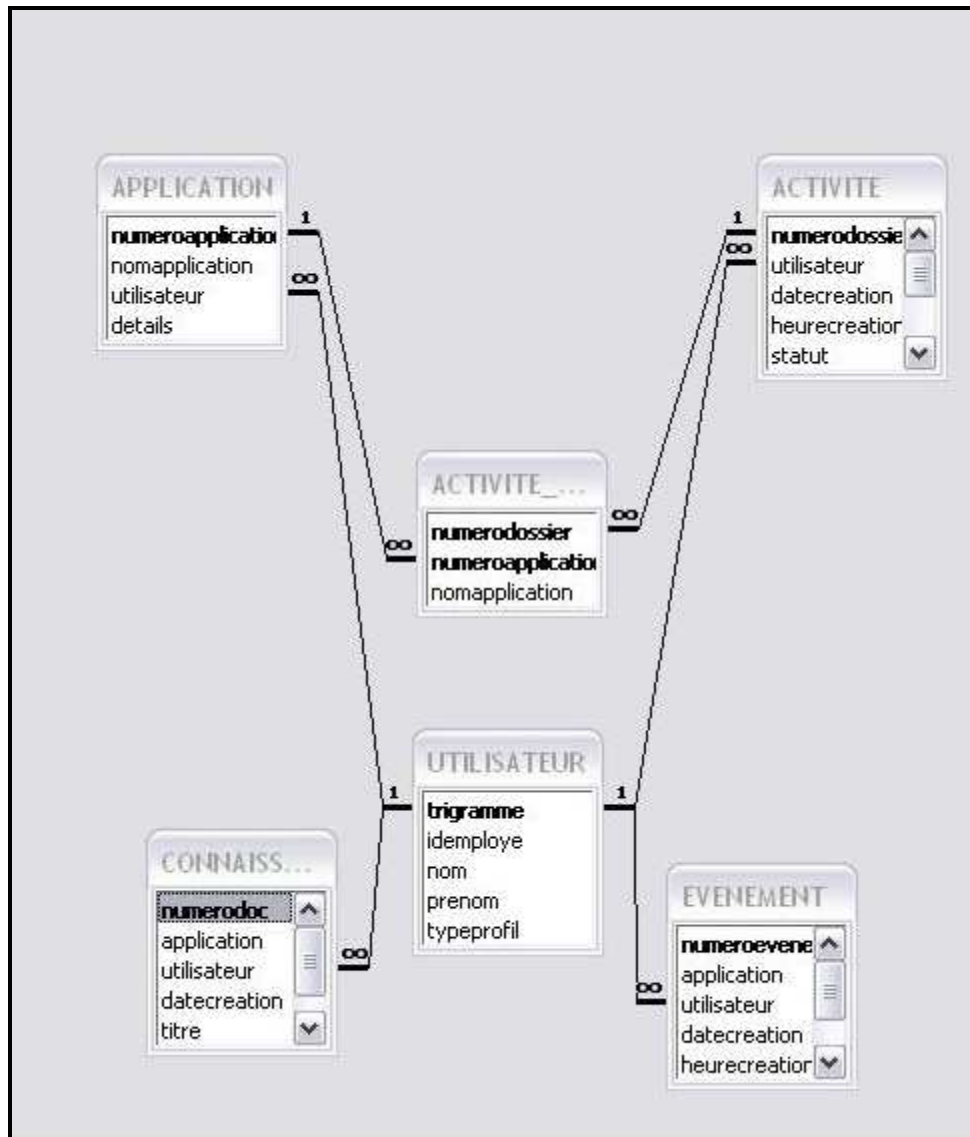


Figure VI.I.VI : Modèle relationnel de la base de données GECKO

Comme défini dans le modèle conceptuel de données, les tables sont reliées par les clés primaires. Les relations définies entre les tables sont :

UTILISATEUR - ACTIVITE : Relation un à plusieurs, signifiant qu'un seul utilisateur peut figurer dans un dossier d'activité, et plusieurs dossier d'activité peuvent contenir un nom d'utilisateur. Les champs liés sont : **trigramme** → **utilisateur**.

UTILISATEUR - APPLICATION : Relation un à plusieurs, signifiant qu'un seul utilisateur peut figurer dans une occurrence de la table APPLICATION, et plusieurs enregistrements de la table APPLICATION peuvent contenir un nom d'utilisateur. Les champs liés sont : **trigramme** → **utilisateur**.

UTILISATEUR - CONNAISSANCE : Relation un à plusieurs, signifiant qu'un seul utilisateur peut figurer dans une occurrence de la table CONNAISSANCE, et plusieurs enregistrements de la table CONNAISSANCE peuvent contenir un nom d'utilisateur. Les champs liés sont : **trigramme** → **utilisateur**.

UTILISATEUR - EVENEMENT : Relation un à plusieurs, signifiant qu'un seul utilisateur peut figurer dans une occurrence de la table EVENEMENT, et plusieurs enregistrements de la table EVENEMENT peuvent contenir un nom d'utilisateur. Les champs liés sont : **trigramme** → **utilisateur**.

2 - La création des requêtes

Une fois les tables achevées, les outils d'extraction de données peuvent être définis. Les requêtes seront construites avec le générateur de requêtes ACCESS 2003 et du langage SQL (Structured Query Language). SQL est un langage performant, utilisé dans les interrogations et les manipulations de données dans les bases de données.

Les clauses utilisées pour rédiger les requêtes seront :

SELECT : Liste des champs qui s'affichent dans le résultat de la requête

FROM : Liste des tables interrogées pour l'extraction des données.

WHERE : Critères de recherche.

GROUP BY : Définition du regroupement.

ORDER BY : Définition d'un tri.

HAVING : Condition de fonction.

Les besoins exprimés par les utilisateurs montrent que les interrogations des données auront principalement lieu sur la même table. C'est un avantage pour la création des requêtes. Les extractions de données exécutées sur une même table ont un délai de restitution de données optimal.

La création des requêtes est effectuée suivant chaque profil d'utilisateur de la base. Deux types de requêtes sont définies : Les requêtes qui seront utilisées par le profil « employé » et celles du profil « chef d'équipe ». Une requête supplémentaire sera rajoutée. Cette requête sera proposée dès le menu d'accueil pour afficher les maintenances prévues le jour même sur les environnements de tests.

Le profil « employé » exécute des requêtes interrogeant les tables suivant des critères qu'il est obligé choisir. . Ainsi une notion de paramètre est introduite dans les requêtes. Pour des questions de performance de la base, les requêtes sans critères ne seront pas proposées. Tous les champs de la table seront affichés par défaut. L'utilisateur a aussi la possibilité de préciser les champs qu'il souhaite afficher.

Le profil « chef d'équipe » en plus d'exécuter les requêtes du profil « employé », a la possibilité d'établir des statistiques sur les tables. Une fonction de comptage des enregistrements est ajoutée aux requêtes déjà existantes et un regroupement selon les critères voulus par l'utilisateur.

Les requêtes de la table ACTIVITE :

Pour effectuer une consultation sur la table activité, six critères de recherche sont définis, donnant ainsi à l'utilisateur la possibilité d'affiner sa recherche. L'utilisateur peut utiliser un ou plusieurs paramètres.

La requête d'interrogation de la table activité propose les paramètres de sélection de données suivants :

- Le champ numerodossier représentant le numéro de dossier.
- Le champ utilisateur représentant le trigramme de l'utilisateur qui a créé le dossier. Afin d'éviter des erreurs de saisie, un menu déroulant propose à l'utilisateur de choisir le trigramme qu'il veut passer en paramètre.
- Le champ application représentant le nom de l'application concernée. Un menu déroulant propose à l'utilisateur de choisir le nom de l'application qu'il veut passer en paramètre.
- Le champ datecreation représentant la date de création du dossier. Un contrôle est effectué sur la valeur saisie par l'utilisateur. La date rentrée en paramètre doit être inférieure ou égale à la date du jour. Un masque de saisie est indiqué à l'utilisateur pour limiter les erreurs de saisie.



Figure VI.II.1 : Fenêtres de saisie du paramètre date avec l'indication du format de date

- Le champ statut représentant le statut du dossier. Un menu déroulant propose à l'utilisateur de choisir le statut qu'il veut passer en paramètre.
- Le champ client représentant le trigramme du client qui a fait la demande de support.

Les données résultant de l'exécution de la requête seront restitués sous forme de tableau (l'affichage classique ACCESS) ou d'états.

Voici un exemple de requête de recherche des dossiers créés par un utilisateur dont le trigramme est un paramètre. Le statut et la période recherche sont également mis en paramètre.

```
PARAMETERS [début période] DateTime, [fin période] DateTime, status Text (6 ), [user] Text (3 );
SELECT ACTIVITE.numerodossier, ACTIVITE.client, ACTIVITE.application, ACTIVITE.titre,
ACTIVITE.application, ACTIVITE.statut, ACTIVITE.utilisateur, ACTIVITE.datecreation
FROM ACTIVITE
WHERE (((ACTIVITE.statut)="status") AND ((ACTIVITE.utilisateur)="user") AND
((ACTIVITE.datecreation) Between "début période" And "fin période"));
```

Les requêtes de la table CONNAISSANCE

Le principe est le même que celui des requêtes de la table ACTIVITE. Les critères de recherche de l'utilisateur sont :

- Le champ numerodocument représentant le numéro du document.
- Le champ application représentant le nom de l'application. Un menu déroulant propose à l'utilisateur de choisir le nom de l'application qu'il veut passer en paramètre.
- Le champ utilisateur représentant le trigramme de l'utilisateur ayant créé l'enregistrement. Un menu déroulant propose à l'utilisateur de choisir le trigramme qu'il veut passer en paramètre.

Les résultats des requêtes seront restitués sous forme de tableau ou d'états.

Voici un exemple de requête de recherche de documentation sur une application dont le nom est en paramètre :

```
PARAMETERS Application Text ( 10 );  
SELECT CONNAISSANCE.numerodoc, CONNAISSANCE.application, CONNAISSANCE.titre,  
CONNAISSANCE.descriptif, CONNAISSANCE.utilisateur  
FROM CONNAISSANCE  
WHERE (((CONNAISSANCE.application)="Application"));
```

Les requêtes de la table APPLICATION

Les critères de recherche proposés à l'utilisateur sont :

- Le champ nomapplication désignant le nom de l'application. Un menu déroulant propose à l'utilisateur de choisir le nom de l'application qu'il veut passer en paramètre.
- Le champ utilisateur désignant le trigramme de la personne ayant créé l'enregistrement. Un menu déroulant propose à l'utilisateur de choisir le trigramme qu'il veut passer en paramètre.

Le nombre total de champs de la table APPLICATION étant de quatre, tous les champs seront affichés dans le résultat des requêtes. Les résultats d'exécution des requêtes seront présentés sous forme de tableau ACCESS ou d'états.

Un exemple de requête est présenté ci - après. Il s'agit d'une recherche d'informations sur une application avec comme critère le nom de l'application et le nom.

```
PARAMETERS [Nom de l'application] Text ( 15 ), Utilisateur Text (3);  
SELECT APPLICATION.numeroapplication, APPLICATION.nomapplication,  
APPLICATION.details, APPLICATION.utilisateur  
FROM APPLICATION  
WHERE (((APPLICATION.nomapplication)="Nom de l'application") AND  
((APPLICATION.utilisateur)="Utilisateur"));
```

Les requêtes de la table EVENEMENT

Les critères de recherche proposés à l'utilisateur sont :

- Le champ application désignant le nom de l'application. Un menu déroulant propose à l'utilisateur de choisir le nom de l'application qu'il veut passer en paramètre.
- Le champ utilisateur désignant le trigramme de la personne ayant créé l'occurrence. Un menu déroulant propose à l'utilisateur de choisir le trigramme qu'il veut passer en paramètre.
- Le champ dateevenement désignant la date à laquelle est planifié l'évènement. Un masque de saisie est indiqué à l'utilisateur pour limiter les erreurs de saisie.

Les résultats d'exécution des requêtes seront présentés sous forme de tableau ACCESS ou d'états. L'exemple ci-dessous présente une requête de recherche d'informations sur une application avec comme critère le nom de l'application et le nom.

```
PARAMETERS Application Text ( 255 ), Utilisateur Text ( 255 ), [Date Evenement] DateTime;  
SELECT EVENEMENT.application, EVENEMENT.titre, EVENEMENT.dateevenement,  
EVENEMENT.details  
FROM EVENEMENT  
WHERE (((EVENEMENT.application)="Application") AND ((EVENEMENT.dateevenement)="Date  
Evenement") AND ((EVENEMENT.utilisateur)="Utilisateur"));
```

Les requêtes spécifiques au profil chef d'équipe :

Les requêtes supplémentaires auxquelles a accès le chef d'équipe sont caractérisées par les fonctions « Count » et « Group By ». La fonction « Count » permet de count le nombre d'occurrences en fonction d'un critère.

L'exemple de la requête ci-dessous permet de compter le nombre de dossier créés par un utilisateur sur une période saisie en paramètre par l'utilisateur. Un contrôle est ajouté sur les dates en paramètre. La date de début doit être inférieure ou égale à la date de fin de période.

```
SELECT Count(" numerodossier") AS Nombre de Dossiers, utilisateur  
FROM ACTIVITE  
WHERE datecreation Between [debut de periode] And [fin de periode]  
GROUP BY ACTIVITE.utilisateur;
```

Ce second exemple porte sur une requête restituant le nombre de dossiers créés par application sur une période rentrée en paramètre par l'utilisateur. Un contrôle est ajouté sur les dates en paramètre. La date de début doit être inférieure ou égale à la date de fin de période.

```
SELECT Count(" numerodossier") AS NBDossier, ACTIVITE.application  
FROM ACTIVITE  
WHERE (((ACTIVITE.datecreation) Between [debut de periode] And [fin de periode]))  
GROUP BY ACTIVITE.application;
```

La requête des évènements du jour :

Afin d'afficher les évènements ou maintenances planifiés dans les environnements de tests au jour de l'interrogation, la requête utilisée aura comme syntaxe :

```
SELECT    EVENEMENT.dateevenement,    EVENEMENT.application,    EVENEMENT.titre,  
EVENEMENT.details  
FROM EVENEMENT  
WHERE (([dateevenement]=Date()));
```

Cette syntaxe signifie :

« Dans la table EVENEMENT, afficher la date de l'évènement, le nom de l'application, le titre de l'évènement et les détails de l'évènement avec la date de l'évènement égale à la date du jour ».

Le résultat de cette requête sera dans un premier temps affiché en actionnant un bouton de commande dans le menu d'accueil de l'utilisateur. Lors de la deuxième phase de mise en production, les évènements du jour seront affichés de manière automatique dans le menu d'accueil à la connexion de l'utilisateur.

3 - Les outils de saisie et les formulaires

Les utilisateurs de GECKO utiliseront des formulaires pour saisir des informations dans la base de données. Il n'y a pas de formulaire spécifique à un profil de connexion. Les formulaires seront identiques quel que soit le profil connecté à la base de données.

Le format de chaque formulaire s'appuie sur les champs de la table à laquelle il fait référence. En plus des champs à remplir, chaque formulaire contient trois fonctions disponibles via un bouton de commandes. La fonction permettant de quitter l'application, la fonction permettant d'imprimer et la fonction permettant de revenir au menu d'accueil.

La fonction « formulaire » est également utilisée pour créer les menus d'accueil de chaque profil. C'est la construction du Menu Général dans Access. Deux menus d'accueil sont définis. L'un pour le profil de connexion « employé » et l'autre pour le profil « chef d'équipe ». Les fonctionnalités sont les mêmes pour les deux menus, à l'exception des statistiques qui ne sont accessibles que par le chef d'équipe. Les fonctions proposées par chaque menu sont gérées par des macros.

Les erreurs de saisie dues à un format de données erronées, sont gérées directement par ACCESS. Le contrôle est effectué par ACCESS sur la base des formats de champs définis dans les tables lors de leur création.

Une fois le développement terminé et l'application prête à fonctionner, le mode création des formulaires sera désactivé afin que les utilisateurs accèdent directement à la mire de saisie de données du formulaire.

Les figures qui suivent montrent un visuel des formulaires créés pour alimenter chaque table. Leur aspect sera par la suite modifié pour améliorer la convivialité de l'interface et adopter des couleurs conformes aux normes HSBC.

3.1 Aperçu du formulaire en création pour la saisie des données dans la table ACTIVITE

The screenshot shows a web form titled "SAISIE DE DOSSIER" for data entry into the "ACTIVITE" table. The form has an orange background and contains several input fields with labels in orange boxes. The fields are arranged in a grid-like fashion. At the bottom right, there are three buttons: "Menu d'accueil", "Impression", and "Quitter".

Label	Input Type
numerodossier:	Text
datecreation:	Text
statut:	Dropdown
utilisateur:	Text
heurecreation:	Text
client:	Text
application:	Text
titre:	Text
resolution:	Text Area

Figure VI.III.I : Formulaire de saisie de la table ACTIVITE

3.2 Aperçu du formulaire en création pour la saisie dans la table CONNAISSANCE

The screenshot shows a web form titled "SAISIE CONNAISSANCE" for data entry into the "CONNAISSANCE" table. The form has an orange background and contains several input fields with labels in orange boxes. The fields are arranged in a grid-like fashion. At the bottom right, there are three buttons: "Menu d'Accueil", "Impression", and "Quitter".

Label	Input Type
numerodoc:	Text
application:	Text
utilisateur:	Text
titre:	Text
datecreation:	Text
descriptif:	Text Area

Figure VI.III.II : Formulaire de saisie de la table CONNAISSANCE

3.3 Aperçu du formulaire en création pour la saisie dans la table APPLICATION

SAISIE APPLICATION

numeroapplication: numeroapplication

nomapplication: nomapplication

utilisateur: utilisateur

details: details

Menu d'Accueil

Imprimer

Quitter

Figure VI.III.III : Formulaire de saisie de la table APPLICATION

3.4 Aperçu du formulaire en création pour la saisie dans la table EVENEMENT

SAISIE D'EVENEMENT

numeroevenement: numeroevenement

application: application

utilisateur: utilisateur

datecreation: datecreation

dateevenement: dateevenement

heurecreation: heurecreation

titre: titre

details: details

Menu d'Accueil

Impression

Quitter

Figure VI.III.IV : Formulaire de saisie de la table EVENEMENT

3.5 Aperçu du formulaire servant à la création du menu d'accueil du profil « chef d'équipe »



Figure VI.III.V : Menu d'accueil du profil « chef d'équipe » dans GECKO

4 - Les Etats et la restitution de données

Les états sont un bon moyen de créer une copie imprimée des informations extraites de la base de données. Ils présentent l'avantage de pouvoir personnaliser le visuel de la présentation des données et de restituer des données plus volumineuses que les formulaires.

Les états seront utilisés pour présenter les données extraites par l'exécution des requêtes. Chaque requête est reliée à un seul état. Il y a autant d'états de créés que de requêtes définies. Ceci quelque soit le profil de l'utilisateur connecté.

Par défaut tous les champs faisant partie du « select » de la requête seront affichés dans l'état. Pour chaque état, une étiquette est créée faisant office de titre de l'état. Cette étiquette est créé dans la partie 'En-tête de l'état'.

Dans la partie détails, les champs sont agencés afin d'améliorer le confort de lecture de l'utilisateur.

La figure ci-après présente l'état 'rapplication' en mode création. Cet état affiche le résultat d'une requête exécutée sur la table APPLICATION.

Figure VI.IV.1 : Etat 'rapplication' en mode création

5 - Les Macros

Les macros sont des fonctions utilisées pour automatiser les différentes tâches dans ACCESS. Dans GECKO, les macros sont utilisées pour l'exécution des tâches proposées par les boutons.

Par exemple, le bouton « Menu d'accueil » présent dans les formulaires fonctionne grâce à une macro, qui sur un clic du bouton ouvre le formulaire « Menu Général ».

L'exécution des fonctionnalités proposées dans le menu d'accueil sont également gérées par des macros.

ACCESS offre deux possibilités de construction de macros : Le générateur de macro d'ACCESS et le générateur de code Visual Basic (une méthode de création de macros par la programmation à l'aide de Visual Basic). Les fonctionnalités de GECKO étant simples à gérer, les macros sont créées à l'aide d'ACCESS.

6 - Les graphiques

Les statistiques obtenues avec l'exécution des requêtes du profil « chef d'équipe » sont également traduites par des graphiques.

La détermination des axes des graphiques se fait avec les champs présents dans le résultat de la requête.

La figure ci-après présente un exemple de graphique créé pour montrer le résultat d'une demande de statistiques sur l'activité de l'équipe du 01/10/2010 au 10/10/2010.

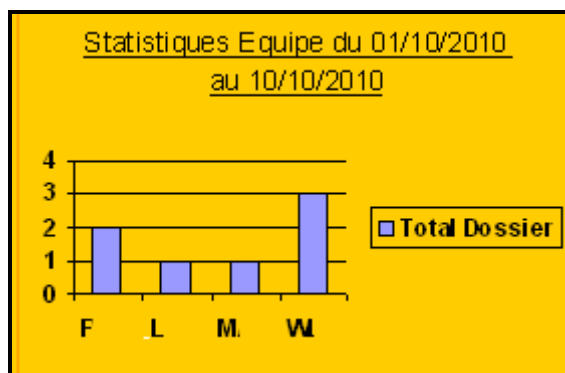


Figure VI.VI.I : Exemple de graphique de statistiques

7 - Le démarrage de l'application et l'interface utilisateur

L'interface utilisateur est l'ensemble des écrans que voit l'utilisateur dans l'application GECKO. Ces écrans sont définis en fonction du profil de connexion. Le premier écran est la mire de connexion de l'utilisateur, suivi du menu d'accueil défini selon le profil utilisateur « chef d'équipe » et « employé ». Les composants des menus sont conformes aux maquettes présentées plus tôt, dans la solution technique.

Afin d'éviter que les utilisateurs ne voient la structure technique de la base de données au démarrage de l'application, une fonction de démarrage est paramétrée dans la base de données. Une fois passée l'étape de connexion, l'utilisateur voit le menu d'accueil de son profil s'afficher. Et non plus les formulaires, les tables...

Cette fonction de démarrage est reliée aux menus utilisateurs précédemment créés. Une barre d'outils est également créée permettant d'avoir accès à toutes les fonctionnalités de GECKO sans avoir à revenir au menu d'accueil à chaque fois. La barre d'outils en construction est représentée par la figure ci-dessous.

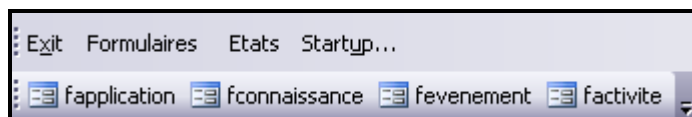


Figure VI.VII.I : Barre d'outils de GECKO

8 - L'analyse de performance de la base de données et la connexion à SQL SERVER 2005

La performance de la base de données sera analysée une fois la structure technique de la base mise en place. Un outil d'analyse de performance est disponible dans ACCESS. Cet outil propose des recommandations d'optimisation des éléments une fois l'analyse terminée. Chaque composant de la base (table, macro, formulaire,...) sera soumis à l'analyseur. Les recommandations proposées par ce dernier seront examinées et appliquées dans les maintenances planifiées de GECKO si elles sont jugées pertinentes.

La connexion de la base de données GECKO au moteur SQL Server 2005, se fera en collaboration avec l'équipe des administrateurs de bases de données de la banque en août 2011. L'équipe Projects and Tests Supports n'ayant pas accès à ce serveur, cette partie du développement ne peut être développée, ni illustrée au moment de la rédaction du mémoire.

9 - La sécurité d'accès à la base de données

La définition des profils ou groupes d'accès à GECKO se fait à l'aide de la fonction Sécurité d'ACCESS. Les figures qui suivent représentent les mires des fonctions permettant de créer des groupes d'utilisateurs et de définir leurs droits dans la base de données.

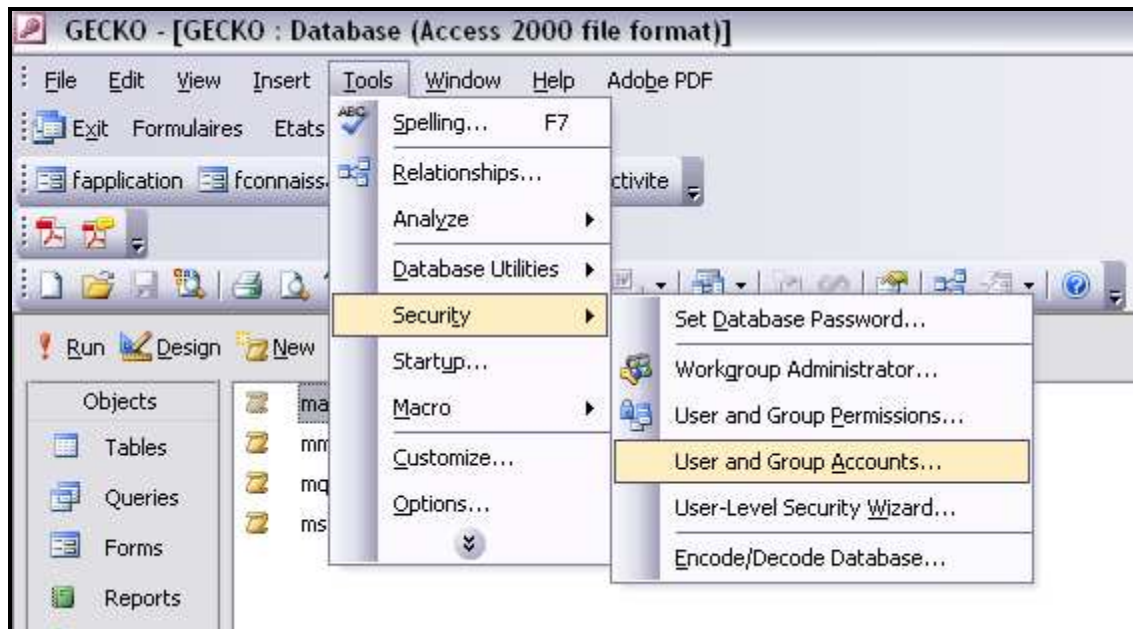


Figure VI.IX.I : Fonction de gestion des droits d'accès des utilisateurs dans ACCESS



Figure VI.IX.II : Fonction de création de groupe d'utilisateurs, de définition d'utilisateur et de gestion de mot de passe



Figure VI.IX.III : Fonction de définition des droits d'accès aux divers éléments de la base de données (tables, formulaires...)

A l'aide de cette fonction, Sécurité, deux groupes d'utilisateurs sont définis : chef d'équipe et employé. Le groupe ADMIN dont fait partie le développeur de la base de données a été automatiquement créé par ACCESS.

Pour gérer la distinction entre les deux profils qui se connectent, le menu d'accueil défini dans la rubrique de création des formulaires, est attribué à chaque groupe correspondant lors de sa création.

Une sécurité supplémentaire est prise en rajoutant les autorisations pour chaque élément de la base de données : Les tables, les formulaires, les requêtes, les états et les macros.

Les accès définis aux tables sont conformes à ceux spécifiés dans la solution technique. A savoir :

- La table UTILISATEUR n'est accessible qu'au groupe ADMIN. Les autres tables sont autorisées en mode lecture, écriture, mise à jour et suppression de données.
- Les formulaires sont accessibles en mode lecture, écriture et mise à jour par les utilisateurs des deux groupes « chef d'équipe » et « employé »
- Les requêtes sont accessibles en mode lecture, écriture et exécution par les utilisateurs des deux groupes « chef d'équipe » et « employé ». Excepté les requêtes de statistiques qui ne sont accessibles qu'en mode lecture et écriture au groupe « chef d'équipe ».
- Les états sont accessibles en mode lecture, écriture et mise à jour par les utilisateurs des deux groupes « chef d'équipe » et « employé »
- L'accès aux macros dépend du profil, car elles sont liées au menu d'accueil de la personne qui se connecte.

10 - La réplication et la méthode de sauvegarde manuelle de la base de données

Une fois le développement achevé et l'accord obtenu pour la livraison de GECKO, une copie de la base de données sera effectuée. Il s'agit d'une copie - maître ou réplique - maître. Cette copie est la seule sur laquelle seront effectuées les modifications de structure de la base de données. ACCESS propose un outil de réplication avec lequel il sera également possible de synchroniser d'autres copies de la base.

La sauvegarde manuelle de la base de données est possible grâce à une fonction d'ACCESS : Back Up Database. Cette option est disponible dans Menu / Utilitaires de bases de données / Sauvegarder la base de données.

La figure ci-dessous donne un aperçu de la fonction de sauvegarde de base de données proposée par ACCESS.

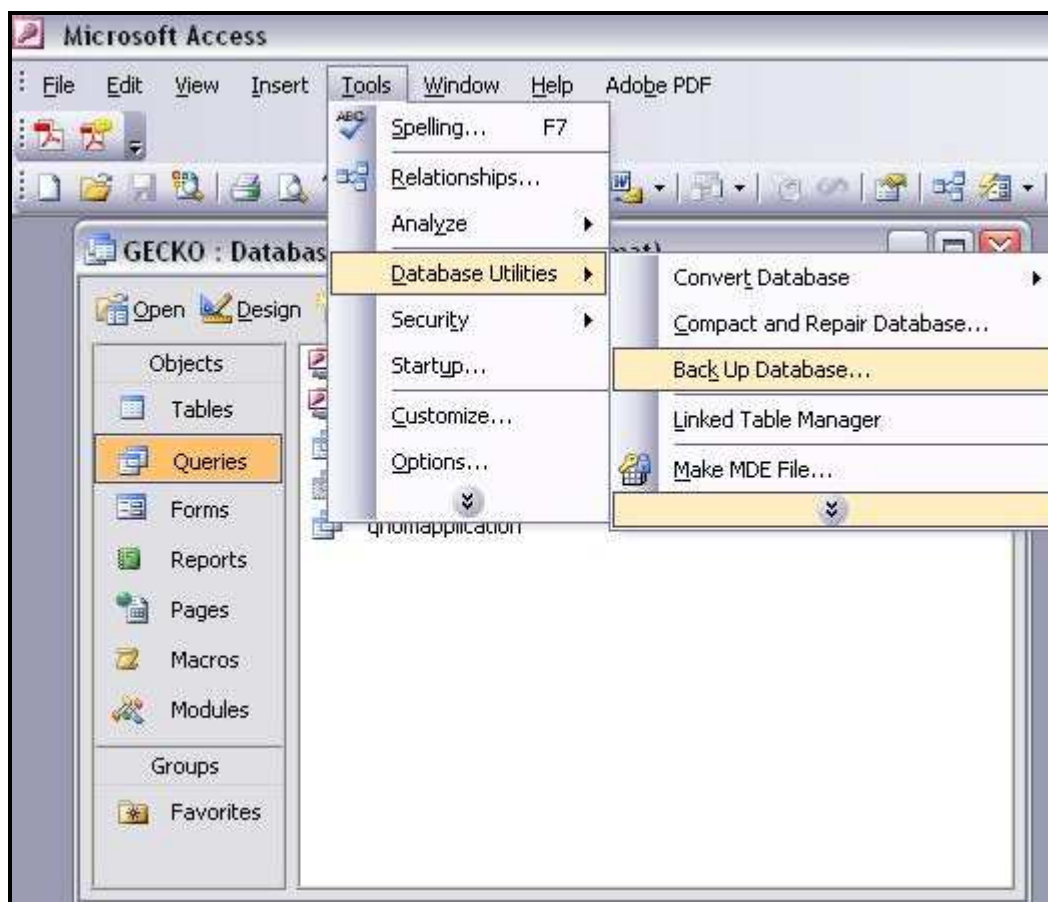


Figure VI.X.I : Fonction de sauvegarde de base de données d'ACCESS 2003

ACCESS propose ensuite un nom pour sous lequel est enregistrée la sauvegarde *base_date de sauvegarde*, exemple : *GECKO_2010-11-15*, et un répertoire où stocker la sauvegarde.

VII - LA PHASE DE TESTS DE GECKO

Les tests représentent une partie importante du projet. La mise en production du logiciel dépend du certificat, TACC (Test Activity Completion Certificate), Certificat de succès de l'activité de test, obtenu à la fin des tests réalisés.

Ce certificat résume l'activité de test réalisée par le projet et montre le ratio et la qualité des tests exécutés.

Le TACC est un document obligatoire de la norme SOX ou LSO en français (des détails supplémentaires sont fournis en annexe). LSO, loi Sarbanes-Oxley est une norme imposée aux projets touchant aux applications financières. La banque est soumise à un audit annuel qui passe en revue les TACC fournis par les projets.

Un TACC est requis à chaque promotion du projet d'un environnement de test au suivant, puis du dernier environnement de test en production.

Les informations nécessaires pour remplir le TACC sont les résultats des tests saisis dans l'outil Hewlett Packard Quality Center.

HP Quality Center est un outil conçu pour l'assurance qualité dans la gestion des tests.

Il permet de :

- Saisir les cas de tests définis dans la stratégie de test du projet.
- Assurer le suivi de l'exécution des tests en mettant à jour leur statut (exécuté, non exécuté, échec, succès ...).
- Gérer les défauts constatés lors des tests et de les assigner soit au chef de projet, soit à la personne en charge de la résolution et/ou du suivi du défaut. Il est ainsi possible d'avoir une évaluation du risque dans la gestion du projet.

1 - Les Rôles et Responsabilités des personnes impliquées dans les phases de test du projet

Les personnes intervenant dans les phases de test du projet sont le Test Manager et les testeurs. Leur rôle se définit comme suit :

Le Test Manager définit la stratégie de test du projet et la rédige. Il saisit les cas de tests dans Quality Center, prépare les dossiers de test pour chaque testeur et organise la réunion d'information du groupe des testeurs. Une fois les tests démarrés, le gestionnaire des tests assure le suivi auprès des testeurs, collecte les informations, les enregistre dans Quality Center et met à jour le statut des tests. Il ouvre les défauts dans Quality Center et les assigne à la personne en charge de la résolution et / ou du suivi.

Le Test Manager est aussi chargé de faire la liaison avec le chef de projet et de lui adresser les problèmes rencontrés et de soulever les risques éventuels pour le projet.

Les testeurs sont en charge d'exécuter les tests demandés dans les dossiers qui leur ont été transmis par le Test Manager, de remplir les informations au fur et à mesure.

Vu les contraintes en termes de ressources du projet GECKO, le collaborateur en charge du projet sera en charge du suivi des tests. Chaque référence au Test Manager sous entendra en fait le membre de l'équipe qui travaille sur GECKO. Il reportera le suivi des incidents au responsable de l'équipe Projects and Tests Environments Support.

2 - La stratégie de test

L'établissement d'une stratégie de test est un élément important du projet. Malgré des besoins clairement définis et un cahier de charges bien rédigé, il subsiste toujours des points dont l'incompréhension ou la mauvaise interprétation apparaissent en cours de projet.

La stratégie de test devra couvrir un maximum de cas de tests en gardant pour objectif :

La possibilité de constater l'avancée du développement de la base de données de manière visuelle : En effet les tests peuvent être interprétés comme un accès au chantier du projet par des personnes autres que les développeurs.

La mise en évidence des différences techniques entre le cahier de charge et le développement : Les interprétations de chaque acteur du projet sont différentes, même si la manière de l'écrire est la même dans le cahier des charges. Par exemple, en demandant une fonctionnalité dans le logiciel comme la présentation des résultats d'une requête, le seul moyen de voir et valider l'ordonnancement des données c'est de le montrer à l'utilisateur.

La correction des anomalies techniques, voire des oublis : Les tests peuvent mettre à jour des anomalies qui n'ont pas été détectées lors du développement ou des fonctions qui ont été tout simplement oubliées.

La clarification d'un point de développement de la base de données : Malgré une définition précise des besoins de l'utilisateur, ce dernier peut changer d'avis en effectuant les tests et demander une modification dans le développement.

La modification du cahier des charges : En fonction des résultats des tests, le cahier des charges peut être modifié et adapté. Au niveau des ressources ou du budget par exemple. La création d'un facteur d'évaluation de risque pour la mise en production du logiciel : Le risque peut être technique, par exemple l'impossibilité de livrer une ou plusieurs fonctionnalité(s), le risque peut également être au niveau du délai de livraison initialement prévu.

La cohérence des cas de test : Chaque cas de test doit être exécutable et pour cela prendre en compte des paramètres tels que l'environnement dans lequel se déroule le test, une description étape par étape du cas de test, le résultat attendu, l'impact en cas d'échec du test... Par exemple un cas de test pertinent doit fournir les informations suivantes au testeur : Comment accéder à la base de données, quel est l'objectif qui est visé, les résultats escomptés. La cohérence et la pertinence des cas de test permettent également d'augmenter la fiabilité du logiciel.

L'historique des tests : En effet chaque cas de test rédigé doit être exécuté et un historique établi. Les cas de tests doivent pouvoir être utilisables à nouveau. Par exemple, pour la correction d'un défaut ou pour le développement d'une nouvelle version du logiciel.

La traçabilité des anomalies, leur suivi et correction : Les cas de tests doivent être construits de telle sorte qu'en cas d'anomalies défauts, ceux-ci puissent être décrits de la manière la plus précise possible par le testeur et donc faciliter le processus de correction.

L'établissement de critères à remplir pour promouvoir le projet à l'étape suivante : La stratégie de test doit définir les pourcentages nécessaires devant être obtenus par les statuts des tests, par les anomalies enregistrées pour permettre au projet de franchir une étape de test. Par exemple, pour une phase d'UAT, les pourcentages tolérés pour la promotion du projet en UAT sont :

- ▶ 100% des tests exécutés et 75% réussis.
- ▶ Toutes les anomalies de Sévérité 1 clôturées.
- ▶ Toutes les anomalies de niveau de Sévérité 2 clôturées.
- ▶ 90% des anomalies de niveau de Sévérité 3 clôturées.

Une communication appropriée entre les différents acteurs du projet : L'exécution des tests permet de fixer des étapes physiques réelles du projet et de pouvoir s'appuyer sur des éléments tangibles pour justifier l'avancée du projet dans les réunions.

La construction de la stratégie de test sera articulée autour du schéma suivant :

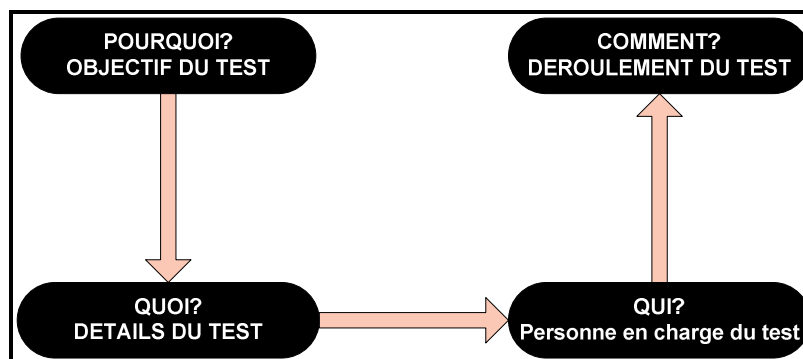


Figure VII.II.I : Logique de définition des cas de tests

Un dossier est également préparé pour chaque testeur par le Test Manager. L'objectif est ici non seulement un suivi de l'activité du testeur, mais également un moyen de communiquer au testeur la finalité des tâches qu'il devra accomplir. Ce dossier contient une fiche pour chaque cas de test. Les fiches contiennent les rubriques suivantes :

L'identification du Projet et des personnes impliquées dans ce cas de test:

- Le nom du projet
- Le nom du Testeur
- Le nom du Chef de projet
- Le nom du Test Manager

L'identification du Test :

- La phase de Test concernée par le cas de test
- Le numéro du Test
- Le nom du cas de Test
- La Date d'exécution planifiée
- La Date d'exécution réelle du test (cette information a son importance, car en cas de retard dans l'exécution des tests, elle peut être invoquée pour soulever un risque de délai de livraison du projet)

Le descriptif du cas de Test :

- L'objectif visé par ce cas de test.
- Les Pré requis pour effectuer le cas de test, par exemple un cas de test doit être exécuté, avant celui qui est décrit par la fiche.
- Les résultats escomptés.

Le descriptif des étapes d'exécution du cas de test : Un maximum de quatre étapes est convenu pour chaque cas de test.

- Le numéro de l'étape.
- Le nom et le descriptif de l'étape.
- Le résultat d'exécution de l'étape (Succès ou Echec).
- Les commentaires du testeur.

Les anomalies relevées lors du test

- Le numéro de l'étape où a été constatée l'anomalie.
- Le descriptif de l'anomalie.
- L'impact de l'anomalie sur le test (aucun impact, bloquant...)

Le résultat du cas de test : regroupe le résultat de toutes les étapes du cas de test, après correction d'éventuelles anomalies et un test de validation de la correction.

Le visa du Test Manager : Une preuve que la fiche a été vue par le Test Manager et les informations consignées dans Quality Center.

Un exemple de fiche de test est présenté à la page suivante.

Tableau 7.2.2 Fiche d'exécution de cas de test

FICHE D'EXECUTION DE TEST									
	Nom du Projet								
	Nom du Testeur								
	Chef de Projet								
	Test Manager								
	Phase de Test								
	Numéro du Test								
	Nom du Cas de Test								
	Date d'exécution planifiée								
	Date d'exécution réelle								
Objectif du test									
Pré requis									
Résultats attendus									
Numéro d'étape	Nom de l'étape et descriptif					Résultat		Commentaires	
Etape 1									
Etape 2									
Etape 3									
Etape 4									
Défauts Relevés									
Numéro d'étape	Descriptif					Impact / Conséquences			
Résultat du cas de test									
Visa du Test Manager									

Deux phases de test seront abordées dans la stratégie de test : La phase de tests unitaires et l'UAT (Users Acceptance Testing).

Du fait des restrictions de temps, de budget et de ressources auxquelles est soumis le projet, la formation des utilisateurs de GECKO sera intégrée à la phase UAT du projet.

2.1 La phase des tests unitaires

L'objectif visé par cette phase est la validation technique des différents éléments de la base de données. En effet à ce stade du projet le squelette de la base de données est terminé. Les tables, formulaires, états et requêtes primaires sont définis et accessibles en lançant le logiciel ACCESS, puis en ouvrant le dossier contenant la base de données. Les droits utilisateurs ne sont pas encore appliqués bien que les profils existent déjà.

Pour les besoins de tests, la base de données sera répliquée en une base GECKO_DIT afin que les saisies tests ne soient pas mélangées aux données réelles qui seront contenues par la base.

Les cas de tests compris dans le scope de cette phase :

Une fois la base de données ouverte dans ACCESS 2003, les tests à exécuter sont :

- L'ouverture de chacune des tables.
- La saisie des données à l'aide des formulaires.
- La consultation des données saisies.
- L'exécution d'une requête avec un et plusieurs critères de recherche (l'utilisateur, l'application ou le numéro de dossier).
- L'affichage d'un état.

Les tests ne faisant pas partie de cette phase :

La phase de tests unitaires n'inclut pas les tests suivants :

- L'accès à la base de données selon un profil avec un mot de passe.
- Les affichages des menus et fonctionnalités correspondant à un profil.
- L'exécution de requêtes de statistiques, par exemple ; quel nombre de dossier ont été créés sur une période lambda.
- L'affichage des événements sur la page d'accueil.

Afin de couvrir les cas de test définis dans le scope de la phase des test unitaires, huit cas de test sont prévus :

Cas de Test numéro 1 : Table ACTIVITE

Pré requis : Aucun

- Ouvrir la table ACTIVITE.
- Saisir un nouveau dossier via le formulaire, essayer de modifier le numéro de dossier.
- Vérifier la présence des données saisies dans la table.
- Exécuter une requête avec un critère de recherche et vérifier la pertinence de l'état affiché.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 2 : Table CONNAISSANCE

Pré requis : Aucun

- Ouvrir la table CONNAISSANCE.
- Saisir un nouveau dossier via le formulaire, essayer de modifier le numéro de dossier.
- Vérifier la présence des données saisies dans la table.
- Exécuter une requête avec un critère de recherche et vérifier la pertinence de l'état affiché.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 3 : Table APPLICATION

Pré requis : Aucun

- Ouvrir la table APPLICATION.
- Vérifier l'affichage des données contenues dans la table.
- Saisir un nouveau dossier, essayer de modifier le numéro de dossier.
- Exécuter une requête avec un critère de recherche et vérifier l'affichage de l'état.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 4 : Table EVENEMENT

Pré requis : Aucun

- Ouvrir la table EVENEMENT.
- Vérifier l'affichage des données contenues dans la table.
- Saisir un nouveau dossier, essayer de modifier le numéro de dossier.
- Exécuter une requête avec un critère de recherche et vérifier la pertinence de l'état affiché.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 5 : Connexion simultanée de tous les testeurs et réexécution des cas de test

Pré requis : Succès des cas de test numéro 1, 2, 3 et 4 et connexion simultanée de tous les testeurs.

- Exécuter le cas de test numéro 1.
- Exécuter le cas de test numéro 2.
- Exécuter le cas de test numéro 3.
- Exécuter le cas de test numéro 4.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 6 : Connexion simultanée de tous les testeurs et exécution de requêtes à plusieurs critères

Pré requis : Succès des cas de test numéro 1, 2, 3 et 4 et connexion simultanée de tous les testeurs.

- Exécuter une requête avec de multiples critères sur la table ACTIVITE.
- Exécuter une requête avec de multiples critères sur la table CONNAISSANCE.
- Exécuter une requête avec de multiples critères sur la table APPLICATION.
- Exécuter une requête avec de multiples critères sur la table EVENEMENT.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 7 : Tentative d'accès à la table UTILISATEUR

Objectif du test : La sécurité de la table UTILISATEUR

Pré requis : Aucun

Résultat attendu : Echec de l'ouverture de la table avec affichage d'un message de refus d'accès.

- Ouvrir la table UTILISATEUR.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de test numéro 8 : Saisie de données au format incompatible avec celui des tables.

Objectif du test : Tester l'intégrité des champs.

Pré requis : Succès des cas de test numéro 1, 2, 3 et 4.

Résultats attendu : Impossibilité de saisir des données non conformes au format pré définis dans la table.

- Effectuer une saisie via le formulaire de la table ACTIVITE avec un format erroné de données. Exemple : changer la date.
- Effectuer une saisie via le formulaire de la table APPLICATION avec un format erroné de données. Exemple : Mettre un nom d'application à la main.
- Effectuer une saisie via le formulaire de la table EVENEMENT avec un format erroné de données. Exemple : Saisir un titre sur 60 caractères.
- Effectuer une saisie via le formulaire de la table CONNAISSANCE avec un format erroné de données. Exemple : Saisir le trigramme d'une personne étrangère à l'équipe.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Les cas de tests décrits ci-dessus sont planifiés sur une durée de soixante minutes. Les tests seront réalisés par la personne de l'équipe en charge du développement et d'un autre collègue pour les tests simultanés.

Les huit fiches d'exécution de test seront identiques pour les deux testeurs.

Les pourcentages ci - dessous détermineront la réussite de la phase de tests unitaires et la condition de promotion du projet en UAT.

- 100% des cas de tests exécutés
- 95% des cas de tests réussis.
- Toutes les anomalies de niveau de Sévérité 1 clôturées.
- Toutes les anomalies de niveau de Sévérité 2 clôturées.
- 90 % des anomalies de niveau Sévérité 3 clôturées.
- 80 % des anomalies de niveau Sévérité 4 clôturées.

2.2 Le Test de validation des utilisateurs (UAT)

L'UAT représente la phase de test où le sponsor du projet ainsi que les personnes ayant été consultées pour la définition des besoins interviennent pour vérifier et valider l'existence et la disponibilité des fonctionnalités demandées au projet. Les tests seront basés sur le fonctionnel et non plus sur le technique comme précédemment dans la phase de tests unitaires.

A ce stade du projet le squelette technique de la base est achevé, les sessions selon le profil de l'utilisateur sont créées, les requêtes complexes et l'affichage des résultats sont disponibles, toutes les fonctions prévues devraient être opérationnelles.

Le logiciel GECKO est accessible à partir d'un exécutable installé sur le poste utilisateur. Pour les besoins de tests, la base de données sera répliquée en une base GECKO_UAT afin que les saisies tests ne soient pas mélangées aux données réelles qui seront contenues par la base.

Cette phase UAT servant également de phase de formation à l'utilisation de GECKO, tous les membres de l'équipe seront impliqués dans les tests. La stratégie de test sera rédigée de telle sorte que chaque profil utilisateur apprenne, teste et maîtrise les fonctions auxquelles il aura accès. Les fiches d'exécution des tests auront alors une double utilité : le suivi des tests et le suivi de la formation des utilisateurs.

Etant donné que GECKO est développé pour une équipe du département informatique, un document expliquant la structure technique de la base de données sera distribué à tous les testeurs lors de la réunion d'information et de préparation de l'UAT.

Les tests compris dans le scope de l'UAT :

Une fois GECKO lancé à partir du poste de chaque testeur, les tests à exécuter couvriront les fonctions suivantes :

- Connectivité selon le profil chef d'équipe ou employé : test du login / mot de passe.
- Affichage des menus d'accueil selon profil.
- Affichage des événements du jour.
- La navigation dans l'application et entre les différentes fonctionnalités.
- Accès et sortie de chaque fonction.
- Fonctions sur chaque table : entrée, sortie, impression, suppression d'enregistrement.
- Impression.
- Consultations et exécutions de requêtes complexes.
- Affichage des résultats des requêtes.
- Requetes de Statistiques, résultats et graphiques.
- Stress Tests : Connexion simultanée de tous les utilisateurs avec exécutions de requêtes lourdes, saisies, impression...

Les tests ne faisant pas partie du scope de l'UAT :

La phase d'UAT n'inclut pas l'affichage dans le menu d'accueil des événements du jour de manière automatique à la connexion de l'utilisateur et les tests de régression de GECKO.

Description des cas de test

Afin de couvrir les tests définis dans le scope de l'UAT, huit cas de test sont prévus :

Cas de Test numéro 1 : Connexion et menus selon les profils utilisateurs.

Objectif : Prise en main de l'application. Valider les mots de passe, l'entrée et la sortie de l'application par les utilisateurs.

Pré requis : Raccourci de l'application installé sur le bureau de l'ordinateur de chaque testeur. Chaque testeur a reçu ses identifiants et mot de passe de connexion à GECKO.

Résultats attendus : Chaque profil a réussi à se connecter, voir les fonctionnalités proposées par le menu d'accueil, à afficher les événements du jour, et à se déconnecter.

- Lancer GECKO par le raccourci.
- Utiliser les identifiants pour se connecter.
- Dans le menu d'accueil, cliquer sur la fonction *AUJOURD'HUI*.
- Sortie du logiciel en utilisant la fonction prévue à cet effet dans le menu d'accueil.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 2 : Utilisation des fonctionnalités de la table ACTIVITE

Objectif : Appréhender et valider les fonctionnalités de la table ACTIVITE

Pré Requis : Succès du cas de test numéro 1

Résultats attendus : Données saisies, consultées et imprimées

- Lancer la fonction ACTIVITE depuis le menu d'accueil.
- Saisir un nouveau dossier et l'imprimer, effacer un dossier.
- Consulter des données et imprimer le résultat de la requête.
- Changer de fonction en passant à une autre table, sortir de l'application.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 3 : Utilisation des fonctionnalités de la table APPLICATION

Objectif : Appréhender et valider les fonctionnalités de la table APPLICATION

Pré Requis : Succès du cas de test numéro 1

Résultats attendus : Données saisies, consultées et imprimées

- Lancer la fonction APPLICATION depuis le menu d'accueil.
- Saisir un nouveau dossier et l'imprimer, effacer un dossier.
- Consulter des données et imprimer le résultat de la requête.
- Changer de fonction en passant à une autre table, sortir de l'application.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 4 : Utilisation des fonctionnalités de la table CONNAISSANCE

Objectif : Appréhender et valider les fonctionnalités de la table CONNAISSANCE

Pré Requis : Succès du cas de test numéro 1

Résultats attendus : Données saisies, consultées et imprimées

- Lancer la fonction CONNAISSANCE depuis le menu d'accueil.
- Saisir un nouveau dossier et l'imprimer, effacer un dossier.
- Consulter des données et imprimer le résultat de la requête.
- Changer de fonction en passant à une autre table, sortir de l'application.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 5 : Utilisation des fonctionnalités de la table EVENEMENT

Objectif : Appréhender et valider les fonctionnalités de la table EVENEMENT

Pré Requis : Succès du cas de test numéro 1

Résultats attendus : Données saisies, consultées et imprimées

- Lancer la fonction EVENEMENT depuis le menu d'accueil.
- Saisir un nouveau dossier et l'imprimer, effacer un dossier.
- Consulter des données et imprimer le résultat de la requête.
- Changer de fonction en passant à une autre table, sortir de l'application.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 6 : Fonctions accessibles au profil CHEF D'EQUIPE

Objectif : Appréhender et valider les résultats de requêtes sur l'activité de l'équipe

Pré Requis : Succès des cas de test 1, 2, 3, 4 et 5, utiliser un profil Chef d'Equipe pour se connecter à GECKO

Résultats attendus : Données extraites et présentées sous forme de tableaux et de graphiques

- Lancer la fonction REQUETES depuis le menu d'accueil
- Exécuter une requête avec plusieurs critères de recherche, contrôler les données du résultat affiché.
- Exécuter une requête avec plusieurs critères dont le comptage par élément, contrôler les données restituées et le graphique du résultat. Exemple : Combien de dossiers ont été créés dans la table ACTIVITE pour l'application BIMAS durant les douze derniers mois. Compter et afficher le résultat par mois.
- Imprimer le résultat des requêtes des deux étapes précédentes.
- Changer de fonction en passant à une autre table, sortir de l'application.
- Remplir la fiche d'exécution de test correspondant à ce cas de test.

Cas de Test numéro 7 : Performances de GECKO

Objectif : Valider les performances de GECKO en termes de temps de réponse lors de la connexion simultanée de tous les utilisateurs

Pré Requis : Succès des cas de test 1, 2, 3, 4, 5 et 6. Connexion de tous les testeurs

Résultats attendus : Temps de réponse optimal

- Lancer l'application et se connecter
- Profil Chef d'équipe : exécuter le cas de test numéro 6
- Profil Testeur : exécuter les cas de tests 2, 3, 4 et 5
- Remplir la fiche d'exécution de test correspondant à ce cas de test

Cas de Test numéro 8 : Données erronées

Objectif : Valider les formats de données définis

Pré Requis : Succès des cas de test 1, 2, 3, 4, 5 et 6.

Résultats attendus : Impossibilité de rentrer des données d'un format exotique

- Exécuter le cas de test numéro 2 en mettant un format de données différent. Par Exemple du texte à la place de chiffres, modifier un trigramme d'utilisateur en y mettant un prénom.
- Exécuter le cas de test numéro 3 en mettant un format de données différent. Par Exemple du texte à la place de chiffres, modifier un trigramme d'utilisateur en y mettant un prénom.
- Exécuter le cas de test numéro 4 en mettant un format de données différent. Par Exemple du texte à la place de chiffres, modifier un trigramme d'utilisateur en y mettant un prénom.

- Exécuter le cas de test numéro 5 en mettant un format de données différent. Par Exemple du texte à la place de chiffres, modifier un trigramme d'utilisateur en y mettant un prénom.
- Exécuter le cas de test numéro 6 en mettant un format de données différent. Par Exemple du texte à la place de chiffres, modifier un trigramme d'utilisateur en y mettant un prénom.
- Remplir la fiche d'exécution de test correspondant à ce cas de test

Les cas de tests décrits de l'UAT sont planifiés sur une durée de huit fois une heure. Afin d'éviter que les tests n'aient un impact sur le travail de l'équipe en terme de disponibilité des ressources, les sept premières sessions d'une heure seront exécutées de manière individuelle. Ce sera également une opportunité pour chaque personne d'avoir du temps pour se familiariser avec GECKO. Seule la dernière session d'une heure est planifiée en groupe pour exécuter le cas de test numéro 7. Ainsi chaque employé devra exécuter les cas de test numéros 1, 2, 3, 4, 5 et 8 individuellement. Le responsable de l'équipe exécutera les cas de test 1, 2, 3, 4, 5, 6 et 8. Les fiches d'exécution des tests, faisant également office de suivi de formation, seront distribuées en fonction.

Toute l'équipe Projects and Tests Environments Support participera aux tests de la phase d'UAT. Au total cinq testeurs avec : quatre profils employé et un profil chef d'équipe.

Les pourcentages ci - dessous détermineront la réussite de la phase d'UAT et la condition de mise en production du projet.

- 100% des cas de tests exécutés
- 95% des cas de tests réussis.
- Toutes les anomalies de niveau de Sévérité 1 clôturées.
- Toutes les anomalies de niveau de Sévérité 2 clôturées.
- 98 % des anomalies de niveau Sévérité 3 clôturées.
- 95 % des anomalies de niveau Sévérité 4 clôturées.

Il est à noter que les résultats de la phase d'UAT, pourront entraîner une modification des délais de livraison ou un fractionnement de livraison des fonctions de GECKO étalé sur l'année 2011. L'objectif du projet reste la livraison au 15 Décembre 2010, d'une version de l'outil rendant possible la saisie de l'activité de l'équipe et la production des statistiques.

2.3 La gestion des tests, des anomalies et de l'assurance qualité

Une fois la stratégie de test rédigée et validée par le sponsor du projet, le Test Manager organise le déroulement des tests, leur suivi et la gestion des anomalies. A la fin de chaque phase de test, le Certificat de garantie de réussite des tests (TACC) est renseigné par le Test Manager. L'outil Quality Center, utilisé pour la gestion des tests, restitue à la fin de chaque étape les pourcentages d'exécution et des statuts des tests.

a. La saisie des cas de tests dans Quality Center

Les cas de test décrits par la stratégie de test sont saisis et planifiés dans Quality center. Pour ce faire, un dossier portant le nom du projet est créé dans Quality Center. Les cas de tests y sont saisis et attribués au testeur concerné. Deux environnements de Quality Center sont utilisés pour la gestion des tests :

TEST PLAN pour la saisie des cas de tests et TEST LAB pour charger et exécuter les jeux de tests et mettre à jour leur statut.

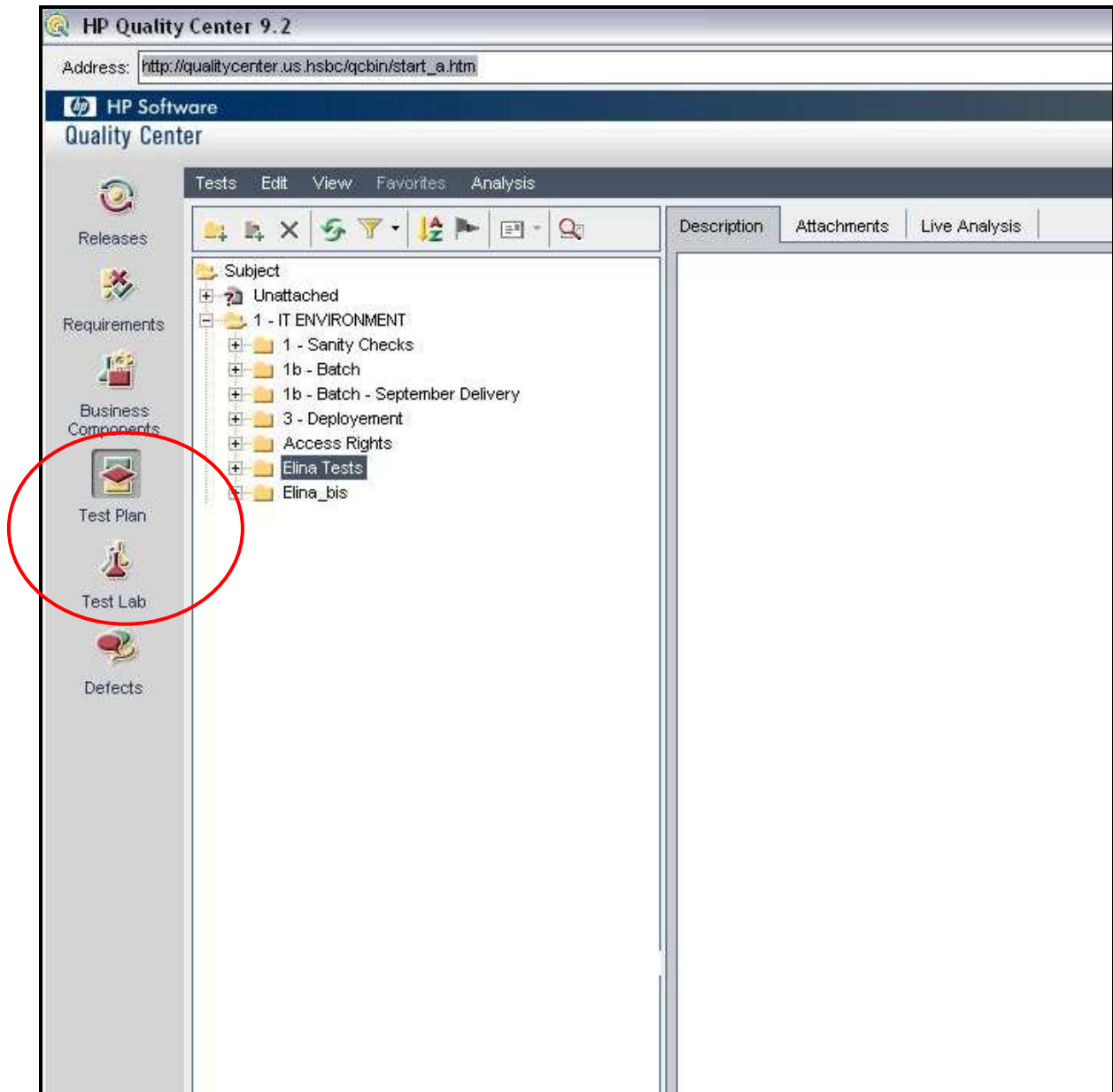


Figure VII.II.III : Vue des rubriques TEST PLAN et TEST LAB de l'outil Hewlett Packard Quality Center

Pour définir un cas de test dans Quality Center, le Test Manager utilise les informations suivantes :

- Le nom du cas test.
- La description globale du cas test.
- Les éventuels commentaires.
- Les étapes du cas de test.
- La description du cas de test.
- Les résultats attendus.
- La personne assignée à l'exécution du test.
- Le nom du programme à tester.
- Le nom du projet.
- L'identifiant sous lequel le projet est enregistré dans la banque.
- Le numéro de Release (au cas où le projet fait partie d'une RDM).
- Le numéro de passe de la Release (au cas où le projet fait partie d'une RDM).
- Le niveau d'importance de l'exécution du cas de test.
- Le risque au cas où les cas de test n'est pas exécuté.
- Le nom de l'application testée.
- La phase de test.

Une fois les cas de test définis, le Test Manager organisera une réunion d'information sur le déroulement de la phase d'UAT en présence de tous les testeurs.

Les informations suivantes seront communiquées lors de la réunion :

- Le calendrier des phases de test unitaires et UAT.
- Les rôles et responsabilités des testeurs, du chef de projet et du Test Manager.
- Les phases de test où chacun interviendra.
- La répartition du temps selon les phases de tests.
- Les fiches d'exécution des tests et leur notice d'utilisation (comment les remplir, quand les rendre au Test Manager).
- La fiche descriptive de la structure de GECKO.
- La phase UAT qui servira de formation.
- Le processus pour obtenir une assistance pendant les tests

b. Le suivi des tests

Une fois les tests démarrés, le Test Manager en assure le suivi. Il s'assure de l'exécution des tests à la date planifiée, collecte les fiches d'exécution des tests, en réalise une synthèse et met à jour le statut des tests dans Quality Center, dans la rubrique Test Lab.

Le statut d'un test peut être succès, échec, non exécuté, bloqué, ou non terminé. En cas d'échec du test, un risque est également rajouté dans Quality Center. Une fois les cas de test chargés dans TEST LAB, le Test Manager rajoute les informations suivantes pour le suivi des tests :

- La date et l'heure planifiées pour l'exécution du cas de test.
- La date et l'heure d'exécution réelles du cas de test.
- Le statut du cas de test.
- Le risque en fonction du statut.

L'évaluation du risque des problèmes rencontrés lors des phases de tests se fait en définissant les conséquences pour le projet au cas où un cas de test n'est pas exécuté. Les paramètres définis dans le tableau ci-dessous donnent le niveau du risque attribué à l'impact causé sur le calendrier de tests du projet.

Tableau 7.2.4 : Différents niveaux de risque du projet et leur définition

Niveau de Risque	Impact
1 – Critique	Le système ne fonctionnera pas dû à un problème de logiciel, de matériel ou de réseau. Impossibilité de continuer les tests. Nécessité de résolution immédiate.
2 – Elevé	Un problème de logiciel, de matériel ou de réseau altère les tests de plusieurs fonctions, certains tests sont malgré tout possible. Il n'y pas de solution alternative. Une résolution est requise aussi vite que possible pour éviter des délais dans le calendrier de test.
3 – Moyen	Un problème de logiciel, de matériel ou de réseau altère les tests d'une fonction, certains tests sont malgré tout possible. Il existe une solution alternative. Doit être résolu, testé à nouveau avant implémentation.
4 – Faible	Un problème mineur ou d'esthétique. Les tests peuvent continuer. Une solution alternative existe si nécessaire. Peut ne pas être résolu avant implémentation.

c. La gestion des défauts

Le Test Manager ouvre des anomalies dans Quality Center pour tous les problèmes rencontrés lors des tests. La rubrique DEFECTS (défauts) est utilisée pour cette tâche.

Chaque anomalie contient un descriptif du problème et est assignée au chef de projet ou à la personne en charge de résoudre le problème. Il existe quatre niveaux de sensibilité d'une anomalie : low, medium, high et critical.

Les anomalies créées pendant les phases de test et leur résolution constituent une base de référence, en cas d'incident lors de la mise en production. En effet l'anomalie créée pour un problème entraîne une attention particulière à la fonction défectueuse, et l'historique de sa résolution est consigné et rapidement accessible si le cas se représente en production.

La figure ci après montre une vue de la rubrique DEFECTS de Quality Center, où sont créées et gérées les anomalies des tests.

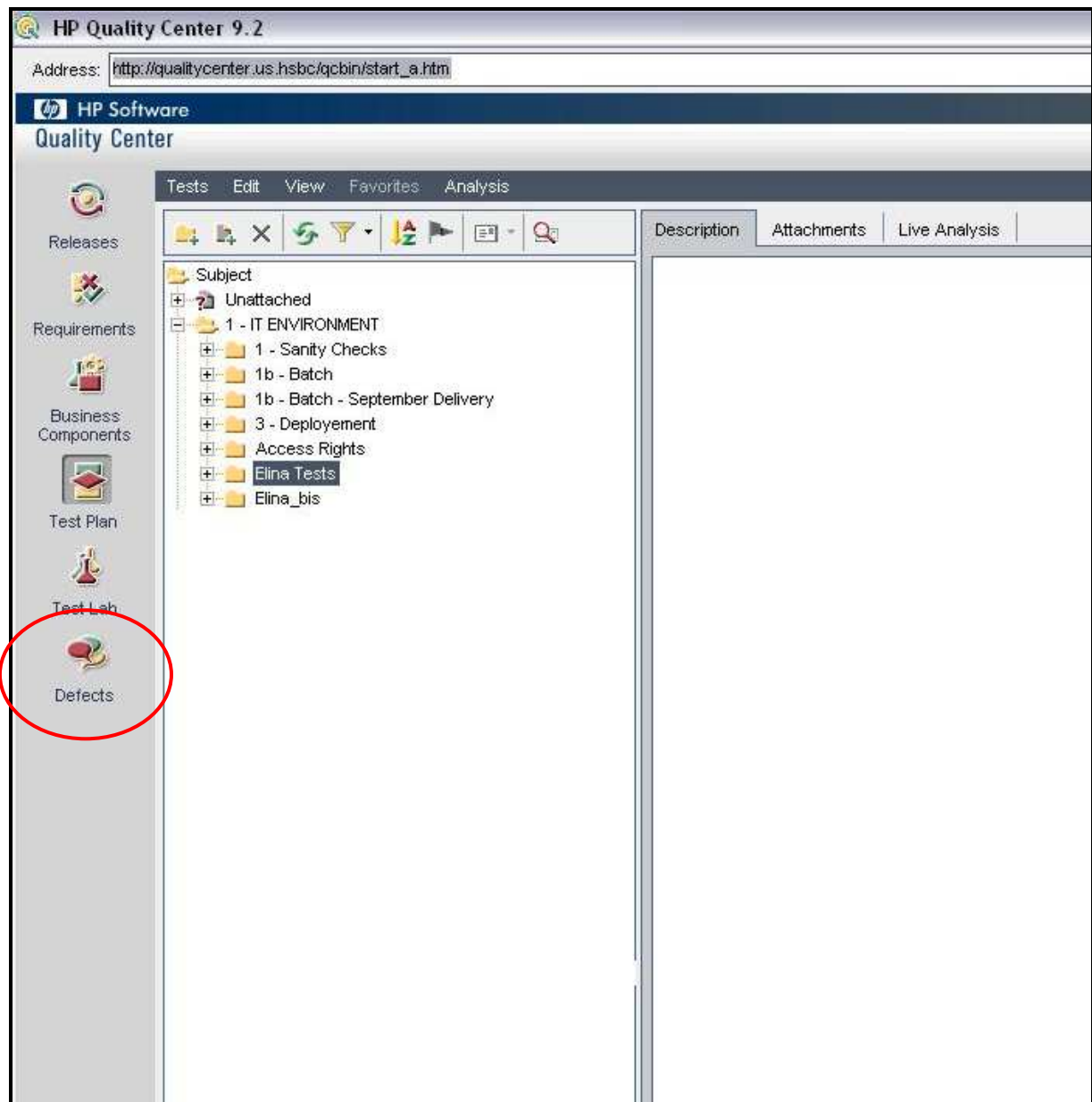


Figure VII.II.V : Vue de la rubrique DEFECTS de l'outil Hewlett Packard Quality Center

Les informations nécessaires au Test Manager pour la création d'une anomalie dans Quality Center sont :

- Le nom de la personne à laquelle assigner le défaut.
- La date à laquelle le problème a été constaté.
- Le niveau de sévérité.
- La phase de test pendant laquelle s'est produit le problème.
- La personne qui a constaté le défaut
- Le niveau du risque
- La priorité du défaut
- Le statut du défaut
- Une description détaillée du problème

Ci-dessous, un exemple d'un formulaire d'ouverture de défaut

Defect: 562 Secom messages NAK due to scrambled data

Details

* Assigned To:

* Raising Group:

Detected in Release: ...

Target Cycle: ...

* Business Domain:

* Detected on Date:

* Business Type:

* Severity: 2- High

* Phase of Origin: 5 - UAT

Detected in Cycle: ...

Error Cause:

Target Release: ...

* Detected By:

* Risk Classification: High

* Priority: 2- High

* Status: Closed

* Detailed Description:

Comments:

Execution Report

Figure VII.II.VI : Formulaire d'ouverture d'une anomalie dans Quality Center

La gestion des anomalies dans Quality Center se fait en suivant le processus défini ci-dessous :

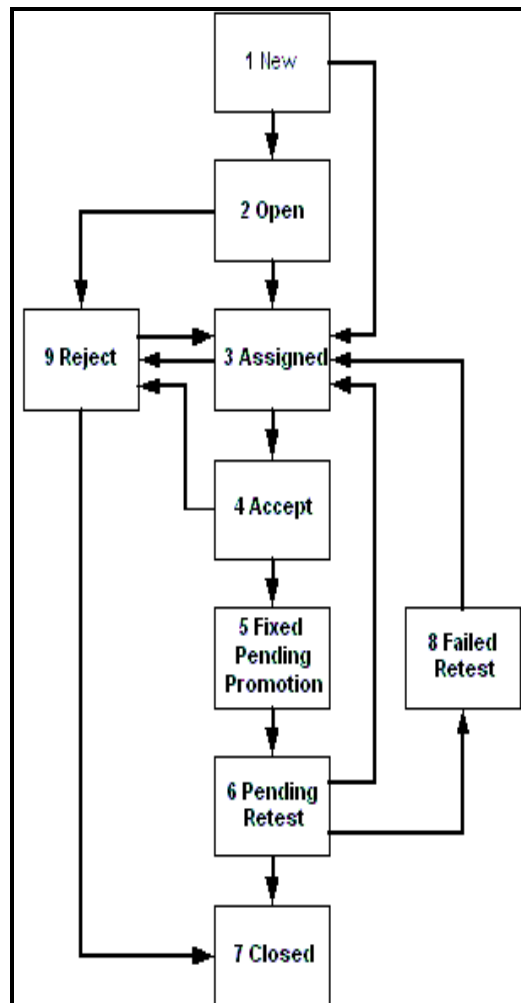


Figure VII.II.VII : Processus de gestion d'une anomalie dans Quality Center

Voici une description de chaque étape de la méthode de gestion des anomalies :

1. **New (Nouveau)** – Lorsqu'une anomalie est créée, son statut est par anomalie mis à nouveau. Les rubriques "Assigned to" and "Team Assigned" sont utilisées pour assigner l'anomalie à la personne en charge du suivi et / ou de la résolution.
2. **Open (Ouvert)** (peut ne pas être utilisé) – Le Test Manager utilise ce statut pour spécifier que le problème désigné par l'anomalie est en cour d'investigation.
3. **Assigned (Assigné)** – Ce statut montre qu'une personne a été assignée à la résolution de l'anomalie. Le Test Manager effectue la première assignation. Si la personne en charge de la résolution doit transférer le problème à quelqu'un d'autre, elle doit rajouter un commentaire expliquant les actions qu'elle a menées et les raisons de l'assignation à un autre collègue. Le Test Manager doit s'assurer que le problème est réellement sous investigation et pas seulement renvoyé de services en services.

- 4. Accepted (Accepté)** (n'est pas toujours utilisé) – La personne assignée à l'anomalie utilise ce statut pour confirmer qu'elle est la bonne personne pour la résolution du problème et s'engage à fournir un historique des actions en cours.

Le tableau ci-dessous présente une définition correspondant au niveau de sévérité attribuée à une anomalie. Ces définitions sont basées sur le dommage que subirait le fonctionnement d'un service si l'anomalie se produit en production.

Tableau 7.2.8 Différents niveaux de sévérité d'une anomalie et leur définition

Niveau de Sevérité	Définition
1 – Critique	Le dysfonctionnement cause une panne du système, une perte de données irrécupérables, un problème de sécurité ou un problème de performance. Le service ne peut fonctionner.
2 – Elevé	Des fonctions critiques du système ne sont pas disponibles à tous les utilisateurs ou certains utilisateurs n'ont accès à aucune fonction. Aucune solution alternative n'existe.
3 – Moyen	Certains utilisateurs n'accèdent pas à des fonctions non critiques du système. Il existe des solution alternatives.
4 – Faible	Un problème d'esthétique ou affectant des fonctions de faible priorité. Des solutions alternatives existent et sont facilement accessibles.

- d. Le Test Activity Completion Certificate (TACC) (un exemplaire est fourni en annexe)

Le TACC est rédigé à la fin de chaque phase de test. Il présente notamment les pourcentages d'exécution des tests et sert de base de décision à la promotion du projet à l'étape suivante. La mise en production n'étant sera autorisée que si le TACC ne contient pas de anomalies avec un niveau critique ou élevé, le Test Manager doit s'assurer de la correction et de la résolution des problèmes de manière pro - active.

VIII - MISE EN PRODUCTION DE GECKO

La mise en production de GECKO initialement planifiée pour le 15 Décembre 2010, sera fractionnée en trois parties. Ce délai est imputé à une surcharge de travail de l'équipe qui retarde le développement de l'application. Les phases de tests UAT sont initialement prévues du 29 Novembre 2010 au 10 Décembre 2010.

1 - Les conditions à remplir pour la mise en production de GECKO

La mise en production de GECKO est soumise à la validation du sponsor du projet. Sa prise de décision est basée sur les documents fournis par le projet attestant de la fiabilité du logiciel, évaluant les risques et les conséquences en cas de problèmes. Ces documents sont :

Le Test Activity Completion Certificate (TACC) : Délivré à la fin de la phase de tests d'UAT par le Test Manager et fournissant une vue des pourcentages d'exécution et de réussite des tests, ainsi que les anomalies ouverts lors des tests.

Le Project Completion Certificate (un formulaire est fourni en annexe): Ce certificat établit une balance entre les besoins exprimés par le sponsor et les réalisations du projet. Il indique si le projet a dépassé le budget alloué, s'il est terminé dans les délais impartis. Les personnes clés ayant donné leur validation au projet après les UAT sont également mentionnées. Le Project Completion Certificate indique aussi si un support est requis après la mise en production.

Le plan de retour arrière en cas de problème lors de la mise en production : C'est un document qui est ajouté au guide de déploiement et décrit la stratégie à suivre, en cas de besoin de retour arrière et les conditions desquelles dépend la décision d'un retour arrière.

La synthèse du suivi des formations des utilisateurs : Ce document fait un récapitulatif du déroulement de la formation des utilisateurs : Les personnes qui ont été formées, les fonctions de GECKO abordées, les difficultés rencontrées, le pourcentage de réussite de la formation, d'éventuelles recommandations du Test Manager.

2 - La préparation de la mise en production

La préparation de la mise en production de GECKO, nécessite trois actions : la rédaction d'un guide de déploiement, d'un plan de retour arrière et l'attribution des droits à tous les utilisateurs selon leur profil d'accès.

2.1 Le guide de déploiement

Le guide de déploiement décrit les étapes de l'installation de GECKO dans le répertoire dédié à héberger la base sur le serveur Windows et sur les postes des utilisateurs. Ce guide concerne uniquement l'étape numéro Un de la livraison de GECKO. Le déploiement sera sous la responsabilité du développeur et de l'administrateur de la base de données.

a. L'hébergement de la base de données sur le serveur Windows

Pré requis : Accès en lecture, écriture et exécution au disque de l'équipe hébergeant le répertoire GECKO

Accès en lecture, écriture et exécution au répertoire du serveur hébergeant les sauvegardes.

Tâches : La base de données GECKO est copiée dans le répertoire GECKO créé sur le disque dédié au département ITQTS.

Une copie de la base nommée GECKO_bkp est sauvegardée sur un autre serveur Windows, spécialement réservé à l'hébergement des sauvegardes.

b. Installation sur les postes utilisateurs

Pré requis : Accès au poste de chaque utilisateur en mode administrateur.

Etape **a** réalisée avec succès.

Tâches : Un raccourci est créé et installé sur le bureau des postes des utilisateurs. Les propriétés du raccourci sont modifiées afin que son visuel soit le suivant :



Figure VIII.II.1 : Logo de l'application GECKO

2.2 La stratégie de retour arrière

La stratégie est construite sur autour de deux points, les conditions auxquelles un retour arrière est effectué et les étapes à suivre.

a. Les évènements entraînant un retour arrière

Un retour arrière de la mise en production de GECKO sera effectué aux conditions suivantes:

- L'utilisation du logiciel cause une panne de l'ordinateur de l'utilisateur.
- L'utilisation du logiciel rend inaccessibles les autres applications.
- Le logiciel est en conflit avec une autre application installé sur le poste.
- Le logiciel ou la base provoque un problème de sécurité des données.
- Tous les utilisateurs n'ont pas accès à GECKO.
- Le logiciel est accessible de manière aléatoire.

b. Les étapes d'un retour arrière

Du fait de la facilité d'installation de l'application sur les postes, un retour arrière se fera en :

- Supprimant le raccourci de connexion à la base de données. Cette tâche sera effectuée par l'utilisateur.
- Fermant l'accès à la base de données aux utilisateurs et en supprimant les droits d'accès au répertoire hébergeant la base. Ces tâches seront exécutées par l'administrateur de la base de données et l'administrateur Windows.

2.3 La définition des droits d'accès des utilisateurs

Une fois la base de donnée GECKO de production prête, l'administrateur de la base de données crée les accès des membres de l'équipe. Un document résumant les détails des utilisateurs et des profils doit lui être adressé par le responsable de l'équipe au plus tard 3 jours avant la date de mise en production. A son tour, l'administrateur doit envoyer les identifiants de connexion et mots de passe à chaque utilisateur la veille du jour de la mise en production.

Tableau 8.2.1 : Formulaire de demande d'accès à GECKO.

DEMANDE D'ACCES A GECKO						
ID EMPLOYE	NOM	PRENOM	TRIGRAMME	PROFIL	DATE DE FIN D'ACCES	COMMENTAIRES
123444785	LAWSON	Elina	WLU	EMPLOYE	31.12.2015	

3 - Les phases de mise en production de GECKO

La livraison de GECKO est planifiée en trois étapes :

3.1 Etape numéro Un : prévue le 15 Décembre 2010 de 12heures à 14 heures

- Les fonctionnalités prévues dans le scope de livraison :
 - L'accès à GECKO par tous les profils.
 - La saisie, la consultation et l'impression des données.
 - L'exécution de requêtes de statistiques, affichage des graphiques.
 - La navigation entre toutes les fonctionnalités.
- Les fonctionnalités non prévues dans le scope de livraison :
 - L'affichage des événements du jour de manière automatique à la connexion de l'utilisateur. Cette fonction sera accessible via une requête disponible dans le menu d'accueil.
 - L'interface Utilisateur développée sous Visual Basic.
 - Le moteur SQL Server 2005.

3.2 Etape numéro Deux : prévue le 18 mars 2011

Les fonctionnalités prévues dans le scope de livraison :

- L'affichage des évènements du jour de manière automatique à la connexion de l'utilisateur.
- L'interface Utilisateur développée sous Visual Basic.

La fonctionnalité non prévue dans le scope de livraison :

- La connexion au moteur SQL Server 2005.

3.3 Etape numéro Trois : prévue le 02 Septembre 2011.

La fonctionnalité prévue dans le scope de livraison :

- La connexion au moteur SQL Server 2005.
- La sauvegarde automatique de la base de données.

Les fonctionnalités non prévue dans le scope de livraison : Aucune.

IX - SUPPORT, MAINTENANCE ET EVOLUTIONS DE GECKO

1 - Le support fourni aux utilisateurs

Une fois GECKO mis en production, le support sera assuré par le développeur de la base de données. Ce dernier pourra demander de l'aide à une personne de l'équipe de d'administration des bases de données de la banque en cas de besoin d'une expertise technique.

Pour chaque problème rencontré lors de l'utilisation du logiciel, une anomalie sera ouverte dans Quality Center. Le suivi et les détails de la résolution (ou pas) du problème y seront également consignés.

Le dysfonctionnement de GECKO, n'ayant pas un impact sévère sur le fonctionnement de l'équipe, la résolution des problèmes liés à la base de données sera gérée en fonction des priorités et des urgences.

2 - La restauration de la base de données en cas de panne ou de corruption de données

La restauration de la base de données se fera en utilisant la sauvegarde la plus récente disponible. La base corrompue sera renommée et déplacée dans un répertoire de maintenance. La sauvegarde sera copiée, collée dans le répertoire habituel de la base et renommée en GECKO.

Le délai estimé pour effectuer la restauration de GECKO est estimé à deux heures. Cette tâche sera sous la responsabilité de la personne de l'équipe Projects and Tests Environments Support en charge de la maintenance de GECKO.

En utilisant la méthode décrite ci-dessus, l'impact pour les utilisateurs de GECKO serait :

- En termes de perte de données, inférieur ou égal à vingt-quatre heures.
- En termes de temps d'indisponibilité de GECKO, inférieur ou égal à huit heures.

3 - Les développements prévus du logiciel GECKO

Des évolutions du logiciel GECKO sont prévues dans les douze mois suivants la livraison. En effet de nouveaux besoins sont apparus au cours du projet, alors que le développement était déjà en cours. Ainsi trois fonctionnalités seront rajoutées :

- La possibilité d'avoir un historique des modifications effectuées sur les occurrences des tables :

A chaque fois qu'un enregistrement sera modifié après sa création, le trigramme de la personne l'ayant modifié et la date de modification seront ajoutées dans un champ historique.

- La possibilité d'avoir des pièces jointes dans les occurrences des tables APPLICATION et CONNAISSANCE :

Une problématique a été soulevée pendant le développement de GECKO : « comment convertir les documents contenant uniquement des schémas de systèmes ou de flux ? »

Ce sont des documents volumineux qui ralentiraient considérablement le temps de restitution des données. La solution envisagée est le rajout d'un champ contenant un lien hypertexte vers un répertoire. Ce répertoire sera dédié au stockage des documents volumineux.

- Un suivi d'activité spécifique à chaque projet sur lequel est dédié un Test Analyst.

Certains projets de la banque sont récurrents de manière annuelle. Ils touchent des applications nécessitant une mise à jour annuelle ou régulière en fonction de l'évolution des lois. Une fiche de suivi de projet présente de nombreux avantages pour l'équipe : La connaissance des environnements de tests nécessaires à un projet, un historique des actions entreprises pour résoudre les problèmes rencontrés par le projet, l'anticipation des besoins du projet, la possibilité pour le Test Analyst de prendre connaissance du projet pour mieux préparer son support.

4 - La méthode employée pour la maintenance et l'évolution de GECKO

La mise en production d'un logiciel ne constitue pas l'étape finale de son cycle de vie. Un suivi et des mises à jour régulières sont indispensables pour garder l'outil de travail performant et adapté aux besoins de ses utilisateurs. Comment maintenir un outil de travail viable, attractif, et éviter qu'il ne devienne obsolète ou rébarbatif à l'utilisation ?

Ce défi est d'autant plus important pour GECKO, qui est un outil développé en interne. Sa maintenance doit être intégrée aux activités de l'équipe et planifiée au même titre que les autres tâches.

Malgré l'importance de la charge de travail de l'équipe, l'ordre de priorité attribué à GECKO ne doit pas être négligeable. Son bon fonctionnement est garant de la restitution de données permettant de quantifier et qualifier l'activité dans l'équipe Projects and Test Environments Support, de justifier l'existence de cette équipe, de montrer leur plus value.

L'observation du mode de travail des membres de l'équipe a permis de construire une stratégie de maintenance de GECKO couvrant une période de douze mois et basée sur trois axes : Un processus de suivi de la vie de GECKO, un plan d'action de la maintenance et la réalisation de la maintenance.

Afin de s'assurer que les fonctions de GECKO resteront adaptées au travail de l'équipe, il est prévu d'intégrer la maintenance de GECKO dans les objectifs annuels fixés aux collaborateurs en début de chaque année. Chaque membre de l'équipe s'occupera à tour de rôle de la maintenance du logiciel sur une période d'une année, en utilisant la méthode de suivi et de maintenance définie plus bas. L'implication des collaborateurs dans cette activité a pour objectif de d'encourager l'utilisation de GECKO par l'équipe.

4.1 Le processus de suivi de GECKO

Le suivi du logiciel consiste à recueillir les informations concernant son fonctionnement durant dix mois, d'en faire une synthèse, puis de la soumettre au responsable de l'équipe afin d'établir les priorités qui feront partie de maintenance de l'année.

La constitution de cet historique se fera de trois manières :

Orale : Cette information peut être recueillie en discutant avec l'utilisateur, lors des réunions d'équipe lorsque les collaborateurs sont invités à donner leurs opinion et propositions.

Par sondage : après les premiers 10 mois d'utilisation, un questionnaire sera soumis aux utilisateurs. Les questions seront préparées en s'appuyant sur les points définis lors de l'expression de besoin dans la phase projet ou écrite. Ce sera également un moyen de mesurer la satisfaction de l'utilisateur et de recueillir ses commentaires et ses besoins.

A l'aide de Quality Center : pour l'historique des problèmes techniques rencontrés, leur résolution ou pas.

4.2 L'établissement du plan d'action de la maintenance

Après dix mois de collecte de données sur le fonctionnement de GECKO, un document de consolidation des informations recueillies sera établi par le collaborateur en charge de la maintenance avec la validation du responsable de l'équipe.

Ce document devra énumérer une liste d'actions à entreprendre pour améliorer l'utilisation du logiciel. Ces actions seront de deux ordres : au niveau des utilisateurs et au niveau de l'application GECKO.

Les actions qui concernent les utilisateurs leurs seront communiquées par le responsable de l'équipe.

Les modifications ou rajout de fonctions dans le logiciel seront rédigées et adressées au développeur en précisant les priorités pour chaque fonction.

Le processus décrit ci-dessus devra prendre un mois.

4.3 L'application de la maintenance

Les modifications techniques de GECKO feront l'objet d'un mini projet d'une durée d'un mois.

X - CONCLUSION

Le support des phases de test des projets de la banque est une activité qui expose l'équipe à une sollicitation permanente de la part de ses clients. Le travail en mode réactif ne laisse pas beaucoup d'opportunités à des initiatives de création ou d'optimisation des outils de travail internes à l'équipe.

Le développement du logiciel GECKO témoigne d'une volonté de changer cette situation : Prendre le temps de créer des indicateurs de l'activité de l'équipe et arrêter la perte de la connaissance acquise.

La mise en production de GECKO, doit être accompagnée d'un engagement de chaque membre de l'équipe à utiliser ce produit, à le maintenir et à le faire évoluer. En d'autres termes, il sera demandé à ses utilisateurs de changer leur manière de travailler actuelle.

Le changement n'étant pas une démarche facile, il faut prévoir du temps pour faire entrer GECKO dans les habitudes de l'équipe. Cependant une fois ce nouveau procédé acquis, chacun se sentira valorisé et reconnu lors la présentation des statistiques de son travail devant ses pairs et sa hiérarchie.

La conception et le développement de l'outil GECKO fut une opportunité de travailler sur des phases de projet différentes de celles dans lesquelles est impliqué habituellement le Test Analyst.

La définition des besoins des utilisateurs, leur formalisation, puis la proposition d'une solution en tenant compte des contraintes furent un défi enrichissant. Les recherches effectuées pour la réalisation de ces étapes ont apporté une meilleure compréhension du cycle de vie des projets, de l'importance de chacune des étapes et de l'investissement humain que demande un projet.

Enfin le développement de la base de données a permis d'acquérir des compétences en méthodologie de gestion de base de données, en langage SQL et en Visual Basic.

La méthode de maintenance prévue pour GECKO sera une occasion de pratiquer et améliorer les compétences nouvellement acquises en gestion de projet, en gestion de base de données et en développement SQL pour l'équipe Projects and Tests Environments Support.

XI - ANNEXES

1 - Loi Sarbanes-Oxley

Finances, gestion, trésorerie

L'objectif de la loi SOX

Adoptée en juillet 2002 par le Congrès américain, la Loi Sarbanes-Oxley (appelée aussi SARBOX ou SOX ou LSO pour les français) oblige les entreprises à répondre de certaines prérogatives administratives dont l'analyse de leurs procédures financières et la publication de leurs résultats dans les plus brefs délais.

L'objectif est clair : il s'agit de restaurer la confiance des investisseurs et de renforcer la gouvernance d'entreprise, largement entamée par les nombreux scandales financiers de 2001 et 2002.

Principe de la loi SOX

Parmi les nombreuses exigences imposées par le Congrès, la LSO vise :

- à garantir une présentation adéquate des rapports financiers,
- à fournir des processus d'alerte,
- à exiger la certification personnelle des rapports financiers par le CEO (PDG) et le CFO (Directeur Financier)
- et également à renforcer les contrôles liés au processus de reporting financier.

Applicabilité de la loi SOX

La LSO s'applique aux sociétés, banques, organismes d'épargne et aux entreprises non américaines qui déposent des dossiers auprès de la SEC sous la section 13(a) ou 15(d) de la loi SEA (Securities Exchange Act) de 1934. Toutes les entreprises cotées sont donc concernées.

Il faut noter deux points importants concernant la loi Sarbanes-Oxley :

- les entreprises non américaines sont aussi concernées,
- les systèmes d'information sont impliqués à double titre par Sarbanes-Oxley
 - dans l'utilisation de l'informatique comme outil de gestion et de contrôle financier
 - dans l'obligation qui instituée (et c'est plus nouveau) d'assurer la sécurité de ce même système informatique.

2 - Test Activity Completion Certificate

TACC registration number and date	<i>For ITQTS Management Only</i>	
Project details		
Project Name:		
Programme HLP:		
Clarity Reference:		
Project Manager:		
Completed activity: (Unit Test, Integration Test, System Test, User Acceptance Test)		
Completed activity leader:		
Activity completion date		
QC Reference		
Domain:	Project:	
Test activity exit status		
Handover criteria	Status	Details
Q1/ All planned tests attempted?	(Yes/No)	Mark "No" when there is no test plan and provide details
Q1.1/ Test plan references for those not attempted		Reference to the QC folder
Q1.2/ Action plan agreed for outstanding tests?	(Yes/No)	
Q1.3/ Action plan detail (<i>append if appropriate</i>)		Reference the action plan or hyperlink to your risk plan
Q2/ Any high risk tests outstanding?	(Yes/No)	
Q2.1/ Test plan references for those outstanding		Reference to the QC folder
Q2.2/ Action plan agreed for outstanding tests?	(Yes/No)	
Q2.3/ Action plan detail (<i>append if appropriate</i>)		Reference the action plan or hyperlink to your risk plan
Q3/ Any severity 1 or 2 defects open/unresolved?	(Yes/No)	
Q3.1/ Defect numbers for those open/unresolved		List defect(s) number(s)
Q3.2/ Action plan agreed for open incidents?	(Yes/No)	
Q3.3/ Action plan detail (<i>append if appropriate</i>)		Reference the action plan or hyperlink to your risk plan
Q4/ Any severity 3 or 4 defects open/unresolved?	(Yes/No)	
Q4.1/ Defect numbers for those open/unresolved		List defect(s) number(s)
Q4.2/ Action plan agreed for open incidents?	(Yes/No)	
Q4.3/ Action plan detail (<i>append if appropriate</i>)		Reference the action plan or hyperlink to your risk plan
PM Quality deliverables (when applicable)		
Q5/ Bimas Checklist reviewed and agreed?	(Y/N or NA)	(PM/PR action)
Q6/ Sybase Checklist reviewed and agreed?	(Y/N or NA)	(PM/PR action)
Q7/ Checkpoint reviewed for all BIMAS deliveries?	(Y/N or NA)	If no, comment the issue. <i>Read more on CHECKPOINT command</i>
Comments		
<i>If applicable insert any additional relevant comments or information here as the release note and QC reference</i>		
Originator	Recipient – Handover Acceptance	
Signature and date of handover	Signature and date of acceptance Full <input type="checkbox"/> or Partial <input type="checkbox"/>	

3 - PROJECT COMPLETION CERTIFICATE

Project details		
Project Name		
Clarity Reference (Project ID)		
Project Start Date and End Date		
Project Closure Type (e.g., <i>Withdrawal, Closed</i>)		
Project Manager		
Project Goal review		
Completion criteria	Status (Yes/No)	Details
Were the objectives met?		
Did the project close within the estimated budget?		
Did the project close within its baselined completion date?		
Project acceptance		
Completion criteria	Status (Yes/No)	Details
Project Sponsor accepted all project results?		
Budget Owner accepted all project results?		
Business As Usual (BAU) Support accepted all project results?		
Project Manager accepted all project results?		
Key Project Stakeholder(s) accept all project results?		
External Customer (if applicable) accepted all project results?		
Is Post Project support required?		
If yes, has Support Agreement / Performance Level Agreement been created and approved? If no, explain.		
Benefits Realized? Yes: Provide detail of projected vs. Actual No: Benefits Tracking Established (Provide detail)		
Has Post Project Review been completed or planned for?		
Has the project been closed out in the tools used to manage the project? (i.e. Clarity, QC, etc.)		
Comments		
Project Manager		Project Sponsor – Full Completion Acceptance
Signature and date of handover		Signature and date of acceptance

XII - BIBLIOGRAPHIE ET REFERENCES INTERNET

1. Livres et ouvrages utilisés durant le projet et pour la rédaction du mémoire

- OFFICE OF GOVERNMENT COMMERCE, 2008. Introduction Officielle à ITIL, Le Cycle de Vie des Services.
- MICRO APPLICATION, 2004. Maîtriser ACCESS 2003.
- COLLECTION REPERE, 2004. ACCESS 2003, Fonctions avancées.
- JOHN L. VIESCAS, 2004. Microsoft Access 2003 au quotidien.
- FERNANDO GUERRERO, 2006. SQL SERVER 2005 étape par étape.
- COLLECTION SOLUTIONS INFORMATIQUES, 2007. ACCESS et SQL SERVER, développez en mode Client - Serveur.
- COLLECTION RESSOURCES INFORMATIQUES, 2007. VBA ACCESS 2007.

2. Liens internet utilisés durant le projet et pour la rédaction du mémoire

- Présentation de Hewlett Packard Quality Center :
https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-127-24_4000_1
- Définition de la loi SOX :
http://www.guideinformatique.com/fiche-sarbanes_oxley-481.htm
- Gestion de projet pour débutant :
<http://www.gestiondeprojet.net/articles/index.htm>
- Modèle Conceptuel de données :
<http://www.commentcamarche.net/contents/merise/mcd.php3>
<http://sqlpro.developpez.com/cours/modelisation/merise/?page=exemple>
- Définition d'une base de données :
<http://www.commentcamarche.net/contents/bdd/bddintro.php3>
- Démarche de construction de BD :
<http://www.ac-grenoble.fr/ecogest/pedago/administration/prodpeda/mcd.htm>
- Définition de la norme ITIL V3 :
<http://www.itil-officialsite.com>

Liste des figures présentées par le mémoire

Titre de la Figure	Page
Figure II.I.I : Modèle de gestion de projet RBPM : Risk Based Project Management	9
Figure II.I.II : Etapes du projet GECKO	10
Figure II.I.III : Etapes du cycle de vie du Service proposées par ITIL	11
Figure II.III.I : Organigramme montrant la structure du département ITQTS dont fait partie l'équipe	13
Figure II.III.II : Fonctionnement du DIT	15
Figure II.III.IV : Mainframe relié aux applications	16
Figure II.III.V : Courbe de la croissance des activités de l'équipe Projects and Tests Environments Support	17
Figure II.III.IV : Structure de l'équipe projet incluant le Test Analyst	18
Figure III.II.I : Un exemple de notes prises par un collaborateur	22
Figure V.I.I : Schéma fonctionnel de GECKO	34
Figure V.I.II : Mire de connexion à GECKO	35
Figure V.I.IV : Menu de la fonction STATISTIQUES	38
Figure V.I.V : Menu de la fonction ACTIVITE	38
Figure V.I.VI : Menu de la fonction DOCUMENTATION	39
Figure V.I.VII : Menu de la fonction EVENEMENT	39
Figure V.I.VIII : Menu de la fonction APPLICATION	40
Figure V.II.VI : Modèle conceptuel de la base de données GECKO	45
Figure V.II.VII : Processus de sauvegarde GECKO : manuel et automatique	46
Figure VI.I.I : Définition de la table ACTIVITE	50
Figure VI.I.II : Définition de la table CONNAISSANCE	51
Figure VI.I.III : Définition de la table APPLICATION	52
Figure VI.I.IV : Définition de la table EVENEMENT	52
Figure VI.I.V : Définition de la table UTILISATEUR	53
Figure VI.I.VI : Modèle relationnel de la base de données GECKO	54
Figure VI.II.I : Fenêtres de saisie du paramètre date avec l'indication du format de date	56
Figure VI.III.I : Formulaire de saisie de la table ACTIVITE	60
Figure VI.III.II : Formulaire de saisie de la table CONNAISSANCE	60
Figure VI.III.III : Formulaire de saisie de la table APPLICATION	61
Figure VI.III.IV : Formulaire de saisie de la table EVENEMENT	61
Figure VI.III.V : Menu d'accueil du profil « chef d'équipe » dans GECKO	62
Figure VI.IV.I : Etat 'rapplication' en mode création	63
Figure VI.VI.I : Exemple de graphique de statistiques	64
Figure VI.VII.I : Barre d'outils de GECKO	64
Figure VI.IX.I : Fonction de gestion des droits d'accès des utilisateurs dans ACCESS	65
Figure VI.IX.II : Fonction de création de groupe d'utilisateurs, de définition d'utilisateur et de gestion de mot de passe	65
Figure VI.IX.III : Fonction de définition des droits d'accès aux divers éléments de la base de données (tables, formulaires...)	66
Figure VI.X.I : Fonction de sauvegarde de base de données d'ACCESS 2003	67
Figure VII.II.I : Logique de définition des cas de tests	71
Figure VII.II.III : Vue des rubriques TEST PLAN et TEST LAB de l'outil Hewlett Packard Quality Center	81
Figure VII.II.V : Vue de la rubrique DEFECTS de l'outil Hewlett Packard Quality Center	84
Figure VII.II.VI : Formulaire d'ouverture d'une anomalie dans Quality Center	85
Figure VII.II.VII : Processus de gestion d'une anomalie dans Quality Center	86
Figure VIII.II.I : Logo de l'application GECKO	90

Liste des tableaux présentés par le mémoire

Titre du Tableau	Page
Tableau 2.3.3 : Environnements de test et nombre d'applications par environnement	15
Tableau 4.1.1 : Fiche de Sondage Numéro Un	26
Tableau 4.1.2 : Fiche de Sondage Numéro Deux	27
Tableau 4.1.3 : Résultat du sondage Numéro un, les caractéristiques du logiciel à concevoir	28
Tableau 4.1.4 : Résultat du sondage Numéro deux, performance de l'équipe et qualité de l'activité	29
Tableau 4.1.5 : Digramme de Gantt du projet GECKO	31
Tableau 5.1.3 : Résumé des fonctionnalités accessibles selon le profil de connexion	36
Tableau 5.2.1 : Descriptif des champs de la table UTILISATEUR	40
Tableau 5.2.2 : Descriptif des champs de la table ACTIVITE	41
Tableau 5.2.3 : Descriptif des champs de la table CONNAISSANCE	42
Tableau 5.2.4 : Descriptif des champs de la table APPLICATION	43
Tableau 5.2.5 : Descriptif des champs de la table EVENEMENT	44
Tableau 5.2.8 : Durées de garde des copies de la base de données GECKO	46
Tableau 7.2.2 : Fiche d'exécution de cas de test	73
Tableau 7.2.4 : Différents niveaux de risque du projet et leur définition	83
Tableau 7.2.8 : Différents niveaux de sévérité d'une anomalie et leur définition	87
Tableau 8.2.1 : Formulaire de demande d'accès à GECKO.	91

Conception et développement d'un logiciel de gestion de l'activité et de la connaissance d'une équipe de support informatique.

Mémoire d'Ingénieur C.N.A.M., Saint Genis Pouilly 2010

RESUME

L'analyse de la méthode de gestion de l'activité et de la connaissance de l'équipe Projects and Tests Environments Support a mis en évidence la nécessité d'un outil de travail dédié à l'équipe.

Les discussions menées avec les membres de l'équipe ont permis de spécifier leurs besoins et les contraintes auxquels sont soumis la conception et le développement d'un tel outil.

Les solutions fonctionnelle et technique ont préconisé une base de données développée sous ACCESS 2003. Une interface simple permettant à la fois d'enregistrer les dossiers traités par l'équipe en un temps minime et de créer une documentation sur la connaissance acquise. L'application est nommée GECKO.

Une stratégie de test de GECKO est mise en place. Elle comprend :

- Les tests unitaires effectués par le développeur de l'application qui valident la structure technique de la base.
- Les tests des utilisateurs finaux qui valident la structure fonctionnelle de GECKO.
- La formation des utilisateurs.
- La gestion des anomalies constatées lors des tests.

L'installation de GECKO sur les postes utilisateurs se fera suivant le guide de déploiement préparé. Une solution de retour arrière est prévue en cas de problèmes de fonctionnement.

Une maintenance de GECKO est prévue annuellement. Des évolutions sont planifiées, notamment la connexion de cette base de données à un moteur SQL Server 2005 et une interface utilisateur améliorée sous Visual Basic 6.5.

Mots Clés : GECKO, logiciel, connaissance, activité, méthode, tests, évolution, maintenance.

SUMMARY

The analysis of the activity and knowledge management methodology used by Projects and Tests Environments Support team, pointed out a need of a specific tool for the team.

Team members' interviews allowed determining their needs and the constraints of the conception and the development of such a tool.

The functional and technical solutions recommended a database developed with ACCESS 2003. A simple interface allowing to record the activity within a short delay and to create a documentation of the knowledge acquired. The software is named GECKO.

A test strategy is put in place. It covers:

- The unit testing performed by the database developer, in order to validate the technical structure of GECKO.
- The final users testing which validate the functional structure of GECKO.
- Users' training.
- Testing defects management.

A deployment guide is set up for GECKO's installation on user's computer. A roll back strategy is planned in case of issues.

An annual maintenance of GECKO is planned. Updates will also be performed, mainly the connection of the database to SQL Server 2005 and an improved user's interface by using Visual Basic 6.5.

Key words: GECKO, software, knowledge, activity, methodology, tests, evolution, maintenance.