

Projet Auto Concept



Adrien Jeser
Mathias Devaux
Michael Vistot
Kélian Bousquet

13/04/2017

Sommaire

<i>I. Contexte</i>	<i>2</i>
<i>II. Méthodologie.....</i>	<i>3</i>
<i>III. Conception de la base de données</i>	<i>5</i>
<i>IV. Projection et limites</i>	<i>7</i>
<i>V. Lexique</i>	<i>9</i>

I. Contexte

L'entreprise CPA (*La Centrale des Pièces Auto*) fait appel à nos services (quatre jeunes étudiants en informatique) afin d'améliorer leur système de gestion actuellement basé sur des fichiers Excel.



Le système actuel comporte de nombreux points faibles à savoir :

- Impossibilité de travailler à plusieurs en même temps sur un document donné
- Incohérences au niveau du stock
- Les fichiers Excel ne respectent pas tous les mêmes standards et conventions de nommage
- Soucis de confidentialité et version de fichier

Avant projet

	Avantages	Inconvénients
Interne	Maitrise de l'outil / Souplesse	Travail collaboratif impossible / Pas de convention de nommage
Externe	Faible coût	Intégrité des données / Sécurité / Obsolescence rapide

Après projet

	Avantages	Inconvénients
Interne	Collaboratif / Intégrité des données	Pas de souplesse / Liaison forte avec les données
Externe	Uniformisation des pratiques / Meilleure vue stratégique et tactique / Meilleure relation client	Nécessite une infrastructure / Collaboration avec un acteur externe / Coût (logiciel et infrastructure) / Formation des utilisateurs

II. Méthodologie

A) Flux de travail

Notre flux de travail s'opère de la conception à la revue par les paires. Il se segmente en trois parties : conception, implémentation, revues.

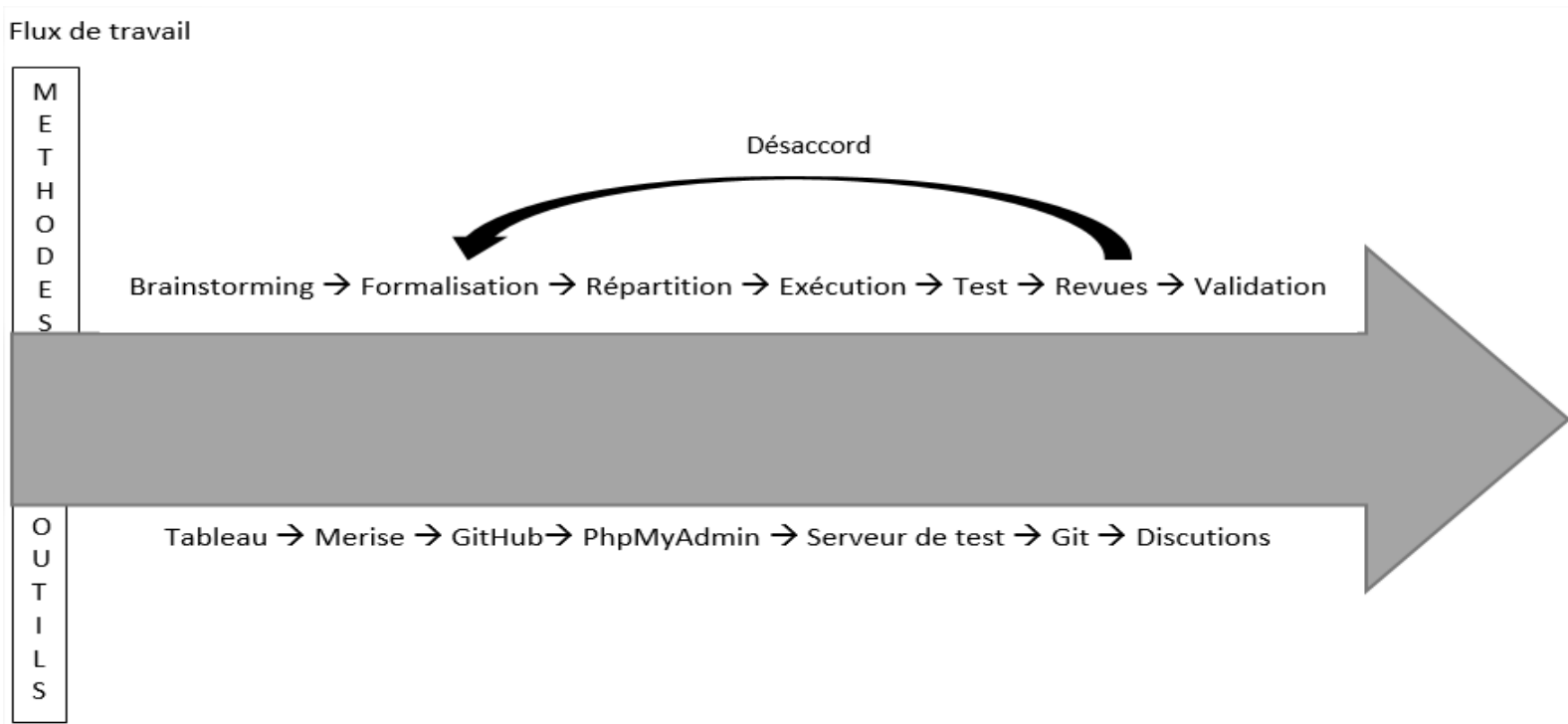


	Tableau	Merise	GitHub	PhpMyAdmin	Serveur de test	Git	Discutions
Avantages	Bon échange	Conceptualisation	Travail collaboratif	Simple d'utilisation	Plateforme de test accessible par tous	Travail indépendant	Intelligence collective
Inconvénients	Demande Animation	Modification incrémentale difficile	Prérequis de formation	Pas d'alternative	Coût de mise en place	Exige de la rigueur	Chronophage

Selon l'avancement d'une tâche, elle s'inscrit dans un modèle organisationnel adapté. Ce qui sous-entend un travail de concert avec un outil précis.

L'établissement du flux au préalable, améliore la collaboration. Pour la conception, nous avons décidé tacitement du modèle. Cette décision favorise le dialogue, autour d'un animateur. Les différentes idées ont été formalisées avec la méthode merise à travers le logiciel Jmerise. Il en résulte un MCD et un MPD.

Afin de se répartir la charge de travail entre les différents collaborateurs, nous avons pris la décision d'utiliser GitHub. Il est intrinsèquement lié à l'outil Git. Ce choix a été capital. Il nous a permis d'éviter la lourdeur du partage de fichiers et ainsi de se concentrer sur les tâches et d'enregistrer des traces de l'avancement du projet. Inconvénient : c'est une dépendance forte, tous les acteurs sont donc obligés de l'utiliser, ce qui peut susciter des réticences.

B) Schéma organisationnel

Dans un premier temps, la méthodologie Agile a été pratiquée. Bien que le Scrum Master ait une place stratégique dans l'assurance du respect de la méthode, dans notre équipe, ce rôle ne fut pas endossé. Ceci affecta le bon fonctionnement du groupe mais une solution à ce problème a été définie de manière collégiale.

Dans un deuxième temps, nous nous sommes réapproprié la méthode en nous dédouanant des aspects bloquants pour le groupe. Chaque individu s'est orienté sur une partie du projet selon ses aspirations, côté technique ou côté conceptuel.

C) Convention de nommage

Pour la création des tables nous nous sommes basés sur la convention suivante :

Type	Nommage	Exemple
Table	Tables	Modele_de_piece
Table de relation	Table_Table2	Partenaire_Adresse
Champs	champs	Nom
Clef Primaire	id	Id
Clef Étrangère	Table_id	Partenaire_id
Réursive	parent_id	parent_id
Vue	Vue_v	Liste_employes_v
Procédure	procedure	nouveau_proprietaire

III. Conception de la base de données

Après analyse du sujet, il est ressorti deux cas d'utilisations au logiciel Excel. Nous les avons implémentés en modules. Pour donner du liant, un troisième module a été créé : Contact.



Module *Contact* :

Regroupe la globalité des personnes (particulier, professionnel, salarié), ainsi que les voitures des contacts. La table partenaire est déterminante car elle permet de discriminer un acteur à savoir :

- Une personne physique
- Un salarié d'une entreprise professionnelle ou un particulier

La création de ce module a permis la généralisation de fournisseurs, clients et salariés au sein de même table.

Module *Facturation* :

L'action de vente enclenche le module facturation. La table bon de commande permet de l'enregistrer. Elle est subdivisée par la table ligne de commande grâce à une cardinalité (1, n).

Le distinguo de son état d'avancement (engagé, livrable, finalisé) est assuré par les champs de date associés.

Le calcul du prix se fait depuis une table de relation regroupant un partenaire, une unité et un modèle de pièce. Par conséquence, on peut comparer les tarifs entre les fournisseurs d'un même modèle de produit. Le prix de vente est obtenu par l'application de la marge désirée sur le prix fournisseur.

En cas de litige, il est répertorié. Si une action corrective est nécessaire, une commande virtuelle sera créée. Exemple d'un retour produit : le client devient donc fournisseur et Autoconcept est client. La pièce litigieuse sera réintroduite dans le stock afin qu'il n'y est jamais d'écart possible. C'est un jeu à somme nulle.

Autre cas de litige, pour une non-conformité des caractéristiques intrinsèques d'un article, le litige est répercuté sur le fournisseur. Un emplacement "rebus" peut être utilisé afin de maintenir l'équilibre du stock par un mouvement vers cet emplacement.

Module **Stock** :

Le cas d'utilisation d'une vente inclut la consultation du module stock à travers trois actions :

- Réception de pièces fournisseur
- Inventaire
- Commandes

Par effet de bord, ils peuvent déclencher un mouvement de stock. La table mouvement les archive. Ceci permet de conserver l'ancienne habitude des coupons « récap ». Chaque produit stocké est identifié par un lot et celui-ci est associé à un emplacement. Ce dernier est récursif pour obtenir une granulométrie variable de localisation d'un lot. De plus, il a un champ booléen indiquant s'il est virtuel ou non. Par exemple, l'emplacement perte qui permet de garder en mémoire un historique des produits perdus.

Chaque produit a un stéréotype de modèle de pièce associé à une catégorie. Il peut avoir une interdépendance avec d'autres, le tout formant un ensemble cohérent.

Un client achetant un polish se voit proposé à la vente le chiffon adéquate à l'application de ce polish.

Un lot est un ensemble d'un même modèle de pièce. Ce concept trace les mouvements de stock. Lors d'une vente, un lot est segmenté et son unité peut être modifiée. Par exemple : le magasinier désire gérer le stock d'un produit par carton alors que la vente de ce produit se fait à l'unité.

IV. Projection et limites

Nous sommes globalement satisfaits du travail proposé, malgré quelques déceptions comme l'omission de la partie service. Celle-ci pourrait être intégrée au prochain livrable s'il a lieu.

Spécifique	Mesurable	Atteignable	Réaliste	Temporellement défini
Apprendre à travailler en collaboration sur une BDD	~ 20 tables ~ 8 vues ~ 3 procédures	Faire un ERP complet	Compétences inégales Manque d'enthousiasme	13 avril

L'objectif de notre équipe était d'apprendre, ce qui supposait de fournir un travail de qualité pour le niveau demandé. Sur ce plan là, nous pensons l'avoir atteint et nous aimerions aller plus loin en développant la partie applicative.

Pendant ce travail, les contraintes auxquelles nous nous sommes confrontées nous ont amené à nous poser quelques questions. Voici le fruit de notre réflexion.

Le choix du SGBD était-il pertinent ?

- Il ne respecte pas le standard SQL 99. Notamment, il ignore les contraintes de vérifications.
- C'est un système de gestion de base de données limité dans ses fonctionnalités. Par exemple, il ne permet pas de réaliser des triggers sur un événement « instand ».
- Mauvaise réputation auprès des professionnels.
- Forte communauté d'utilisateurs.

L'intelligence doit-elle être dans la base de données ou dans l'application ?

Tout dépend de la volonté du concepteur du logiciel. Ce dernier a sa vision et ses préférences :

- S'il veut garantir l'intégrité des données, il choisira de mettre l'intelligence dans la base de données. Par contre, les modifications d'implémentation sont très coûteuses ;
- Par opposition, en choisissant l'intelligence dans l'application, il aura une souplesse dans l'implémentation mais lui seul sera gage de l'intégrité des données.

Comment garantir la sécurité des identifiants MySQL dans l'application ?

Plusieurs solutions sont possibles :

- Faire l'autruche (ne rien faire) ;
- Demander un mot de passe au lancement de l'application afin de déchiffrer les identifiants dans un fichier ;
- Faire un serveur d'authentification ;
- Utiliser le système de droit déjà intégré dans MySQL.

V. Lexique

BDD : Base De Données

MCD : Modèle Conceptuel Données

MPD : Modèle Physique Données