

***13/04/2017***

***Projet Auto Concept***

***Adrien Jeser***

***Mathias Devaux***

***Michael Vistot***

***Kélian Bousquet***

***Sommaire***

[***I.*** ***Contexte*** 2](#_Toc479695237)

[***II.*** ***Méthodologie*** 3](#_Toc479695238)

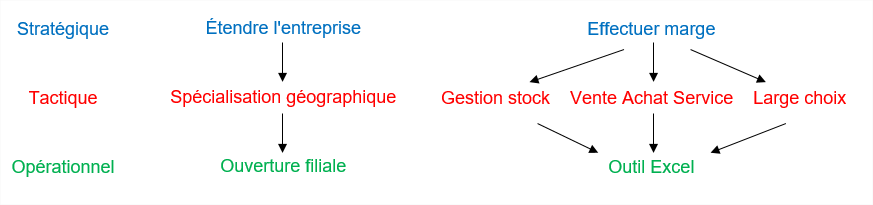
[***III.*** ***Conception de la base de données*** 5](#_Toc479695239)

[***IV.*** ***Projection et limites*** 7](#_Toc479695240)

[***V.*** ***Lexique***](#_Toc479695242) 9

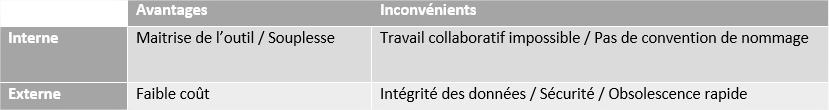
1. ***Contexte***

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

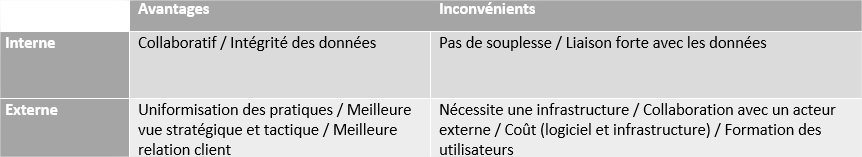


Le système actuel comporte de nombreux points faibles à savoir :

* 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

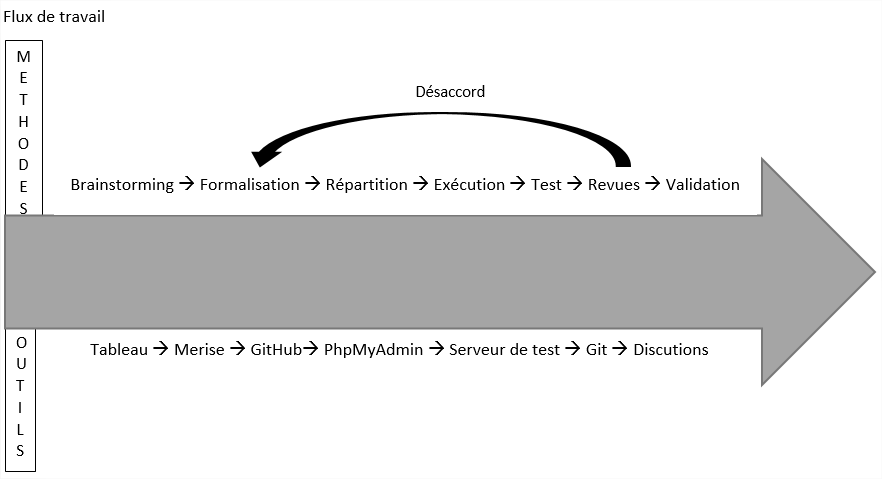
Avant projet

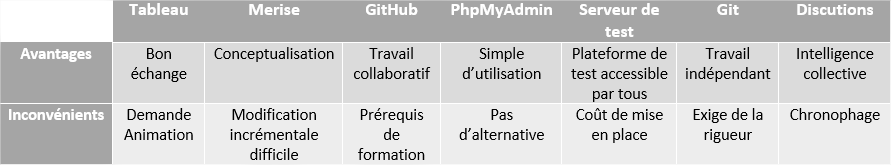
Après projet



1. ***Méthodologie***
2. ***Flux de travail***

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





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 formalisé 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’utilisé 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 fichier et ainsi de se concentrer sur les taches 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.

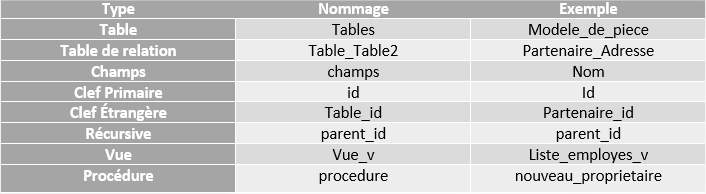
1. ***Schéma organisationnel***

La méthodologie agile a été pratiqué dans un premier temps. Le Scrum master à une place stratégique dans l’assurance du respect de la méthode. Dans notre équipe ce rôle ne fut pas endossé ce qui amputa son bon fonctionnement. Le problème a été identifié de manière collégiale, donc nous nous sommes réapproprié la méthode en ce dédouanant des aspects bloquant pour le groupe.

Chaque individu c’est orienté sur une partie du projet selon ses aspirations côté technique ou plutôt côté conceptuel.

1. ***Convention de nommage***

Pour la création des tables nous nous somme basé sur la convention suivante :



1. ***Conception de la base de données***

Après analyse du sujet il en est ressorti deux cas d’utilisations au logiciel Excel. Que nous avons implémenté en module. 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
* Une ou des adresses

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) selon 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 fournisseur d’un même modèle de produit. Le prix de vente est obtenu avec la somme du prix fournisseur et l’application de la marge désirée.

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

Module **Stock :**

Le cas d’utilisation d’une vente inclus 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 archives ce qui permet de conserver l’ancienne habitude des coupons « récap ». Chaque produit stocké est identifié par un lot et lui-même par 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. Par exemple l’emplacement perte qui permet de garder en mémoire un historique des produits perdu.

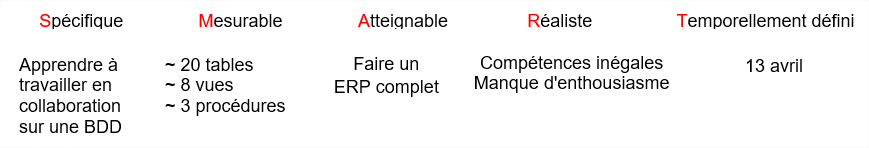
Chaque produit a un stéréotype de modèle de pièce associé à une catégorie. Il peut avoir une interdépendance avec d’autres. Le tout formant un ensemble cohérant.

Un client s’attend que pour l’achat d’un polich le chiffon correspondant soit proposé à la vente.

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 par unité

1. ***Projection et limites***

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



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 pouvoir voir plus loin en développent la partie applicative.

Par voie de faits les contraintes auquel nous nous sommes confrontées nous ont amené à se poser quelques questions.

**Le choix du SGBD était-il pertinent ?**

1. Il ne respecte pas le standard SQL 99. Notamment, il ignore les contraintes de vérifications
2. C’est une base de données limitée dans ses fonctionnalités.

Par exemple : il ne permet pas de réaliser des triggers sur un événement « instand »

1. Mauvaise réputation auprès des professionnels

**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 à sa vision et ses préférences :

Soit il veut l’intégrité des données garantie, il choisira de mettre l’intelligence dans la base de données par contre les modifications d’implémentation sont très couteuses.

Par opposition, il aura une souplesse dans l’implémentation mais lui seul sera gage de l’intégrité des données en choisissant l’intelligence dans l’application.

**Comment garantir que 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

1. ***Lexique***

BDD : Base De Données

MCD : Modèle Conceptuel Données

MPD : Modèle Physique Données