Catalogue Manager

**2015-2017**

2015-2017



Sommaire

[1 Contexte professionnel](#_Toc462299627) 3

[2 Présentation de l’application](#_Toc462299627) 3

3 Schéma d'architecture 4

[4](#_Toc462299630) Méthodologie de travail 5

4.1 Etude du besoin..……………………………………………………………………………………………………….5

4.2 Gestion de projet.………………………………………………………………………………………………………5

4.3 Développement……………………………………………………………………………………………….……..6

4.4 Structure de l'application…………………………………………………………………………………………..9

# Contexte professionnel

Cette situation professionnelle est réalisée dans le carde de l’entreprise INFOTEL, société de conseil spécialisée dans le développement informatique, support et relationnel client. Dans le cadre de l’amélioration de la productivité des différents plateaux en mode « catalogue », nous avons été chargé de réaliser l’application Catalogue Manager.

# Présentation de l’application

Catalogue Manager est une application web destinée à la gestion de plateau, au travers d’un catalogue de services. Cette application va donc gérer les différentes entités ci-dessous :

* un catalogue de services avec ses différentes lignes de services couplées à des niveaux de complexité ;
* des commandes associées au catalogue de service ;
* des tickets associés aux commandes passées par le client ;
* une facturation au travers d’une consolidation mensuelle des données ;
* la gestion des KPIs, qui sont les indicateurs clé de performance (Key Performance Indicator), tout au long du cycle de vie des tickets.

Du fait de l’implication du client dans la gestion des commandes, l’application sera utilisée non seulement en interne au sein d’Infotel mais aussi via les sites clients.

# Schéma d’architecture

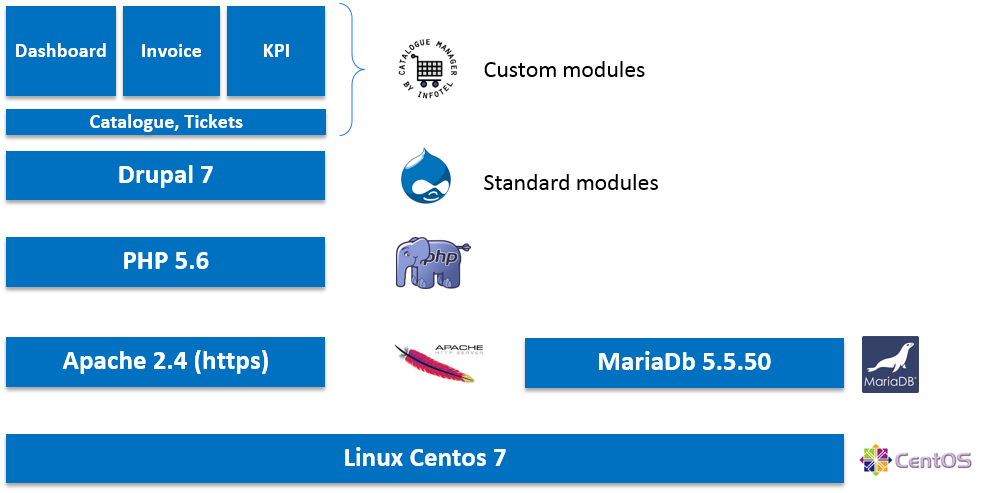
L’application Catalogue Manager est une distribution de **Drupal 7**.

**Drupal** est un serveur de CMS, permettant de gérer des contenus au travers d’une application web. C’est un un outil hors pair qui permet à ses utilisateurs de créer des sites internet complexes et sur-mesure, et répondre ainsi à des cahiers des charges des clients les plus exigeants. Le CMS reste tout de même programmable et personnalisable à souhait afin de répondre au mieux aux besoins

**LAMP** (Linux - Apache – MariaDb - PHP)

Les rôles de ces quatre composants sont les suivants :

* **Linux** assure l'attribution des ressources aux autres composants (Rôle d'un Système d'exploitation ou OS pour Operating System) ;
* **Apache** est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur) ;
* **MariaDb** est un système de gestion de bases de données (**SGBD**). Il permet de stocker et d'organiser des données ;
* le langage de script PHP permet la génération de pages web dynamiques et la communication avec le serveur MariaDb.



# Méthodologie de travail

## Etude du besoin

Dès mon arrivé chez INFOTEL Conseil le 05 Octobre 2015, j’ai été affecté au projet de Catalogue Manager. L’application avait été demandée par un plateau de l’entreprise, mais rien n’avait été commencé, seulement un document avait été rédigé énumérant vaguement les différentes fonctionnalités demandées. J’ai donc dû dans un premier temps aller démarcher les chefs de projets de différents plateaux afin de leur poser des questions et rédiger le BRD (Buisines Requirement Dossier) qui donnait une liste exhaustive des fonctionnalités demandées. Etant donné que ces besoins recueillis venaient de trois plateaux différents, j’ai dû, à l’aide de l’équipe avant-vente d’Infotel, trier et choisir les fonctionnalités que l’on embarquerait dans notre première version de l’application Catalogue Manager. Suite à cela, j’ai dû rédiger de la documentation concernant Catalogue Manager : le user case, un document résumant les propositions de module, et enfin un schéma résumant le fonctionnement de l’application.

## Gestion de projet

Le projet de l’application Catalogue Manager est géré grâce à la méthode agile Scrum. Cette méthode a été choisie principalement car elle nous permet beaucoup plus de flexibilité qu’une méthode de développement classique, mais aussi car le client (Product Owner) était présent dans nos bureaux, et c’est un principe fondamental à l’application de la méthode Scrum.

L’organisation de notre équipe est simple, et suit les règles établies par la méthode Scrum. Chaque matin, nous effectuons le « stand-up meeting » ou nous définissons avec toutes l’équipe de développement et le Scrum Master (qui fait aussi office de responsable technique) : ce que nous avons fait le jour dernier, ce que nous comptons faire à ce jour et pour finir, les points bloquants que nous avons rencontrés. Lors de notre première réunion, nous avons dû définir nos sprints, la durée de ceux-ci, et les tickets que nous emporterions dedans. Nous avons défini des sprints de 4 semaines. A la fin d’un sprint, nous devons livrer une version stable et fonctionnelle au Product Owner (client).

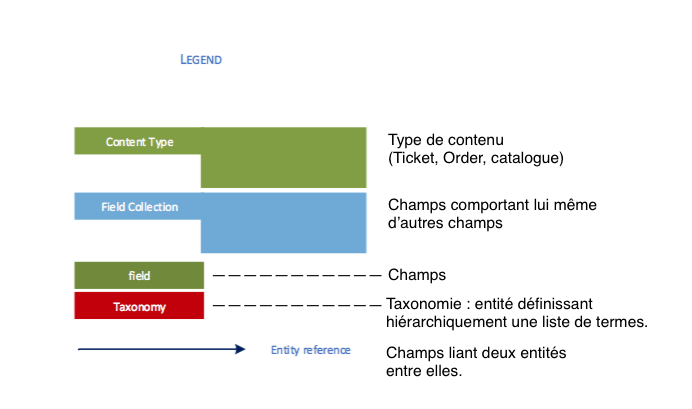
De plus, nous utilisons une application de gestion de projet « JIRA ». Pour chaque demande, évolution, ou modification demandées nous créons un ticket sur celui-ci, lui assignons un titre, un court texte décrivant la tâche à appliquer, un statut de priorité (majeur, mineur, prioritaire, …) ainsi qu’un score de « point d’effort » (voir annexe 1).

## Développement

Nous avons décidé de développer la solution avec Drupal 7, car ce CMS répondait aux besoins. Suite à une période de formation, nous avons commencé à développer l’applications avec les fonctionnalités que Drupal nous proposait. Malgré les aprioris vis à vis des CMS actuellement, Drupal regorge de possibilités, fonctionnalité, et nécessite donc un apprentissage sur la durée.

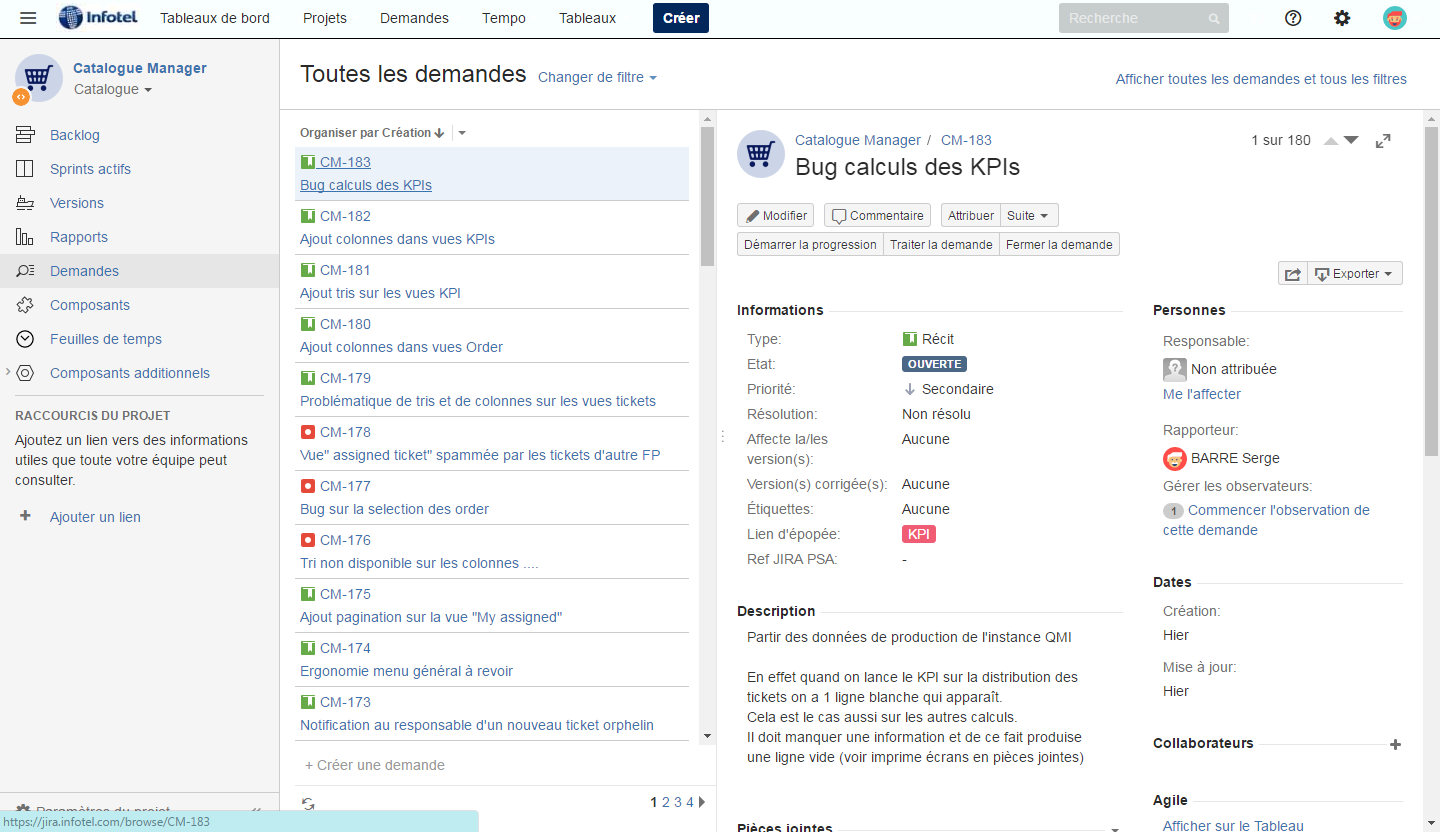
## Structure de l’application

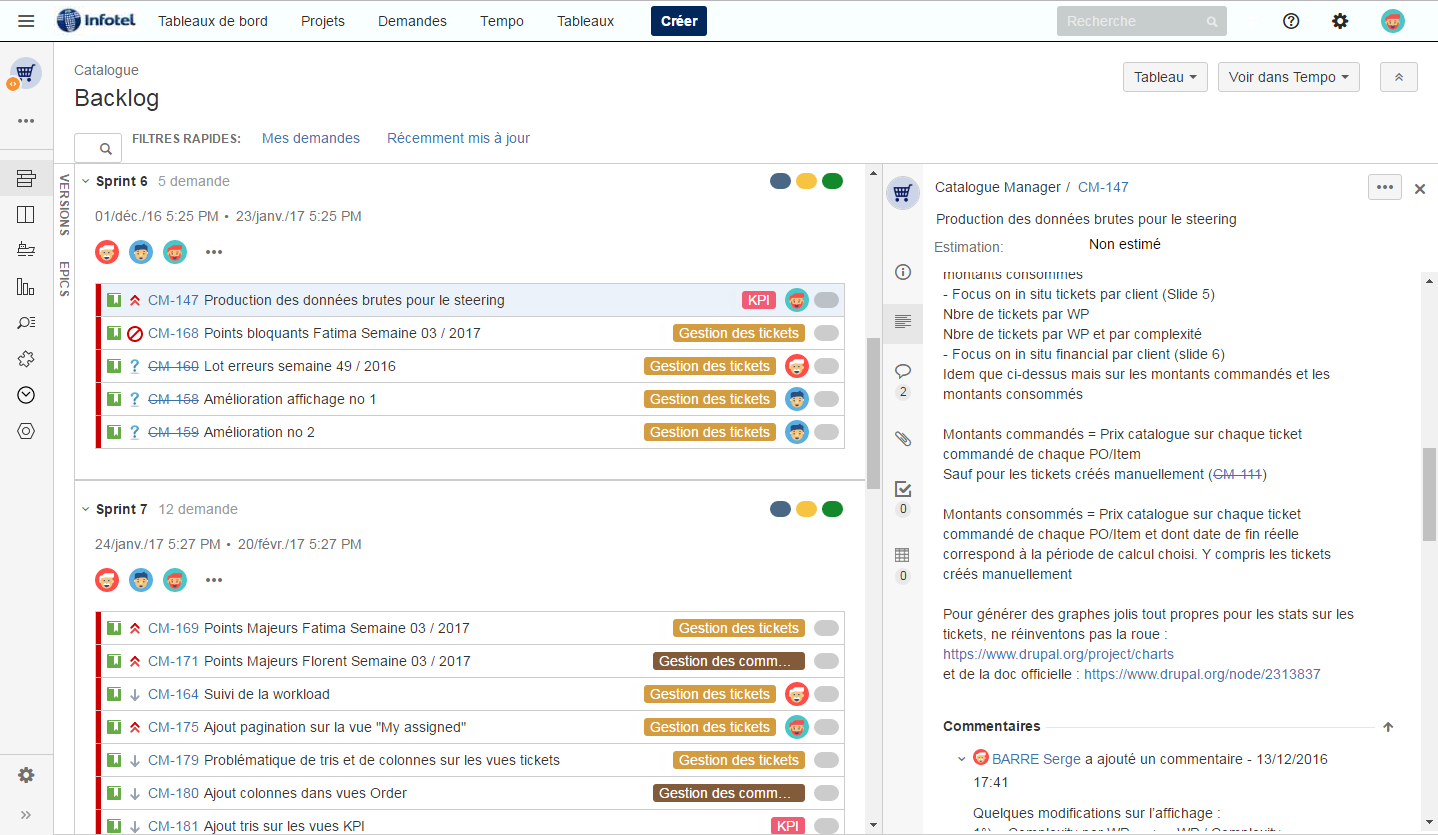
## 

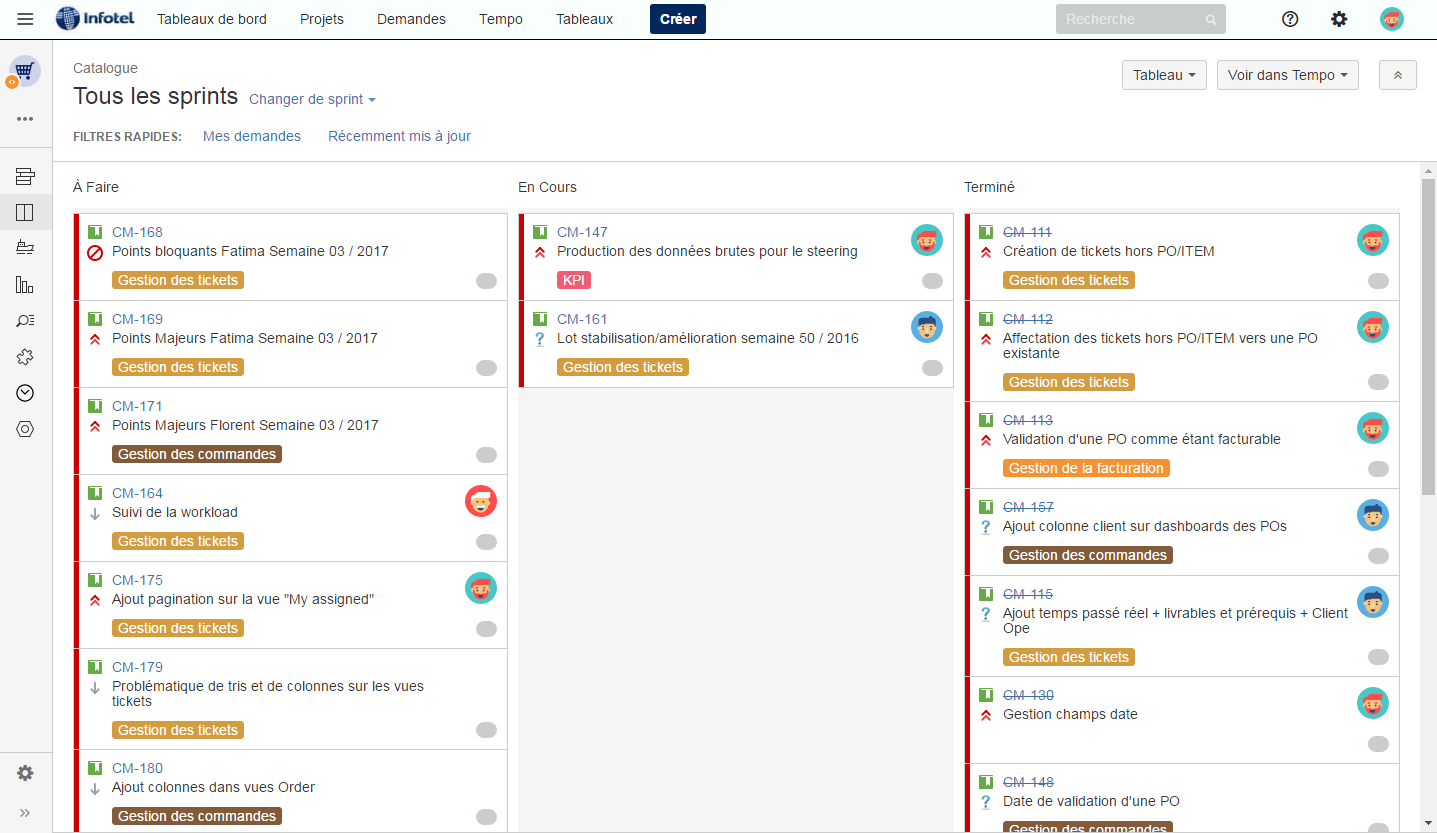
****

**Structure de Catalogue Manager**

**Annexe 1 : Capture d’écran JIRA**

**Toutes les demandes JIRA**

**  
Répartition des demandes en différents sprints JIRA**

**Dashboard récapitulant le statut des tickets JIRA**

## Image%20JIRA/JIRA-modificationTicket.pngAjout/Modification d’un ticket JIRA