

INSTITUT PAUL LAMBIN

BAC 2 INFORMATIQUE DE GESTION

CAHIER DES CHARGES

---

# Outil de gestion des bugs et des demandes d'améliorations

---

*Auteurs :*

Christopher SACRÉ  
Antoine MANIET  
Alexandre MANIET  
Timothé BOUVIN  
Damien MEUR

*Professeur :*

B. LEHMANN

4 décembre 2016

# Table des matières

<b>1</b>	<b>Contexte</b>	<b>2</b>
<b>2</b>	<b>Analyse de l'existant</b>	<b>2</b>
2.1	Acteurs . . . . .	2
2.2	Documents manipulés . . . . .	3
2.2.1	Fiches descriptive et signalétique . . . . .	3
2.2.2	E-mail . . . . .	3
2.3	Règles de gestion . . . . .	4
2.3.1	Détection du bug . . . . .	4
2.3.2	Gestion des commentaires . . . . .	4
2.3.3	Modification des documents . . . . .	4
2.3.4	Subdivision en différents états . . . . .	4
2.3.5	Finalisation du traitement . . . . .	5
<b>3</b>	<b>Objet du marché</b>	<b>5</b>
3.0.1	Objectifs . . . . .	5
3.0.2	Contraintes . . . . .	5
<b>4</b>	<b>Spécifications applicatives</b>	<b>6</b>
4.1	L'application <a href="#">SoftRepair</a> <sup>TM</sup> . . . . .	6
4.2	Description des fonctionnalités . . . . .	6
4.3	Description du travail des utilisateurs . . . . .	8
4.3.1	L'intermédiaire . . . . .	8
4.3.2	Le commercial . . . . .	8
4.3.3	Le membre d'équipe de développement . . . . .	8
4.3.4	Le responsable d'équipe de développement . . . . .	9
4.4	Description des données . . . . .	12
<b>5</b>	<b>Glossaire</b>	<b>13</b>

# 1 Contexte

La société *SoftDevelop S.A.* développe trois lignes de produits-logiciels (*Soft Clients*, *Soft Fournisseurs* et *Soft Compta*) vendus par des intermédiaires. Chaque intermédiaire a un contact commercial chez SoftDevelop.

C'est notamment via cet intermédiaire commercial que le client prend contact avec l'entreprise elle-même. La prise de contact peut avoir différentes raisons : demande d'amélioration des produits, correction de bugs ou encore demande d'information. C'est le commercial qui fait ensuite parvenir les demandes aux équipes de développement.

De ce fait, la société *SoftDevelop S.A.* souhaite sous-traiter la création d'un nouveau programme leur permettant d'automatiser la gestion des demandes d'améliorations et des bugs. Ce cahier des charges a pour but de présenter la solution proposée par notre entreprise.

# 2 Analyse de l'existant

## 2.1 Acteurs

Lors de l'analyse du système actuel de *SoftDevelop S.A.*, 3 acteurs se sont démarqués pour la gestion des bugs.

**Le commercial :** il fait le lien entre l'équipe de développement et le client.

C'est via le commercial que le client fait savoir qu'un bug est apparu.

**Le membre de l'équipe :** il s'agit de la personne qui s'occupera de corriger le bug, de réaliser des tests.

**Le responsable :** il supervise plusieurs membres d'équipes et s'occupe de vérifier si le travail est correctement effectué et si le bug est corrigé.

Une autre dénomination revient fréquemment : **l'utilisateur**, cette dernière est utilisée désigner les 3 acteurs cités précédemment en même temps et sans faire de distinction au niveau des rôles.

## 2.2 Documents manipulés

Actuellement, la transmission des bugs se fait via e-mail et à l'aide d'une fiche signalétique et d'une fiche descriptive.

### 2.2.1 Fiches descriptive et signalétique

Exemple fiche descriptive	Exemple fiche signalétique
<p>* Hardware : Nom de la station: Description PC : Version OS:</p> <p>* Utilisateur a-t-il les droits d'administrateur sur le PC : OUI - NON</p> <p>* Paramètres régionaux :</p> <p>* Que s'est-il passé : ** Comportement : ** Message d'erreur :</p> <p>* Reproductibilité : * Toujours / parfois / quelques fois / expérimenté une seule fois * Commentaires:</p> <p>* Etapes détaillées pour reproduire le bug :</p> <p>* Remarques éventuelles :</p>	<p>*Logiciel : **Type de logiciel : Soft Clients / Soft Fournisseur / Soft Compta **Version du logiciel : **Version corrigée :</p> <p>*Nom intermédiaire :</p> <p>*Priorité : *P1 / P2 / P3 *Commentaires :</p> <p>*Sévérité : *Critique / Majeure / Modérée / Mineure *Commentaires :</p> <p>*Résumé :</p>

TABLE 1 – Exemple de fiches descriptive et signalétique

Explication des différentes priorités :

- **P1** : la priorité la plus haute, réservée pour les demandes urgentes.
- **P2** : la priorité moyenne.
- **P3** : la priorité la plus basse.

Explication des différentes sévérités :

- **Critique** : la sévérité la plus haute ; elle indique que le bug est bloquant et empêche le fonctionnement du logiciel et donc la production chez les clients.
- **Majeure** : elle indique que le bug est grave mais qu'il existe une façon de travailler pour le contourner.
- **Modérée** : elle indique que le bug n'empêche pas le logiciel de fonctionner ni le client de travailler.
- **Mineure** : il s'agit d'un bug « cosmétique » qui sera résolu en dernier lieu.

#### 2.2.2 E-mail

Par ailleurs, tous les commentaires se font actuellement via e-mail. C'est d'ailleurs via ce même e-mail que l'on suivra l'évolution du bug et que l'on

gardera une trace de ce de dernier.

## 2.3 Règles de gestion

### 2.3.1 Détection du bug

Le bug est détecté / l'amélioration est demandée le plus souvent par les utilisateurs qui font parvenir cette information au commercial avec qui il est en contact. C'est le commercial qui s'occupe d'encoder le bug / l'amélioration dans le système et c'est donc ce dernier qui lance le processus de gestion.

### 2.3.2 Gestion des commentaires

Actuellement les commentaires sont gérés par e-mail. Tous les commentaires concernant un bug se rattachant au mail signalant le bug en question. De plus, l'évolution du projet se fait également via e-mail.

### 2.3.3 Modification des documents

À tout moment un membre d'équipe ou un responsable peut modifier *la fiche signalétique du bug* :

- le logiciel sur lequel il a rencontré le bug
- le nom de l'intermédiaire
- la version du logiciel dans laquelle le bug est survenu
- la version du logiciel dans laquelle le bug a été corrigé
- la priorité
- la sévérité
- le résumé

Il faudra donc permettre la modification de ces informations par le responsable et ses membres d'équipes.

Par opposition, *la fiche descriptive d'un bug* ne pourra jamais être modifiée. Dans le cas où cette dernière serait erronée ou contiendrait une description peu fiable ou devrait être tout simplement modifiée, cela se fera à l'aide de commentaires.

### 2.3.4 Subdivision en différents états

La gestion d'un bug<sup>1</sup> se fait à l'aide de différents états, un bug ayant toujours un état à un moment donné :

- Nouveau
- Confirmé
- Pris en charge
- Corrigé
- Vérifié

---

1. Nous utiliserons le mot « bug » pour se référer à un bug ou à une demande d'amélioration RFE dans le reste de ce cahier des charges.

- Clos
- Rejeté
- Non-reproductible
- Doublon
- Rouvert

Le passage d'un état à l'autre symbolise un changement dans la façon de traiter ce bug et son avancée dans le processus de résolution. Il faudra donc notifier les utilisateurs de l'évolution du projet via ces différents états.<sup>2</sup>

### 2.3.5 Finalisation du traitement

Seul le responsable peut décider si un bug est corrigé ou non. C'est ce dernier qui prendra la décision finale et qui mettra fin au traitement en livrant la version corrigée au client. Par ailleurs un bug pourra très bien être rouvert par la suite.

## 3 Objet du marché

La société *SoftDevelop S.A.* souhaite sous-traiter le développement d'un outil de gestion des bugs et des demandes d'améliorations, pour ses logiciels.

### 3.0.1 Objectifs

- Signaler un bug ou RFE
- Faire évoluer le bug ou le RFE tout au long de son cycle de vie
- Rechercher un bug ou un RFE sur base de son identifiant unique ou de toute information de sa fiche signalétique
- Visualiser l'ensemble des bugs ou RFE correspondant aux critères de recherche, sous un format de tableau, chaque ligne correspondant à un bug
- Livrer un mode d'emploi avec l'outil de gestion des bugs

### 3.0.2 Contraintes

- La maintenance de l'outil de gestion des bugs et des demandes d'améliorations sera assurée pendant trois ans
- L'application ne sera développée qu'en utilisant des standards ouverts du marché tant au niveau de la programmation que du langage de programmation

---

2. Si un bug clos veut être rouvert, on ne changera pas l'état de ce dernier, mais on considèrera qu'il s'agit d'un nouveau bug.

## 4 Spécifications applicatives

### 4.1 L'application **SoftRepair**<sup>TM</sup>

TO DO = INTRODUCTION

### 4.2 Description des fonctionnalités

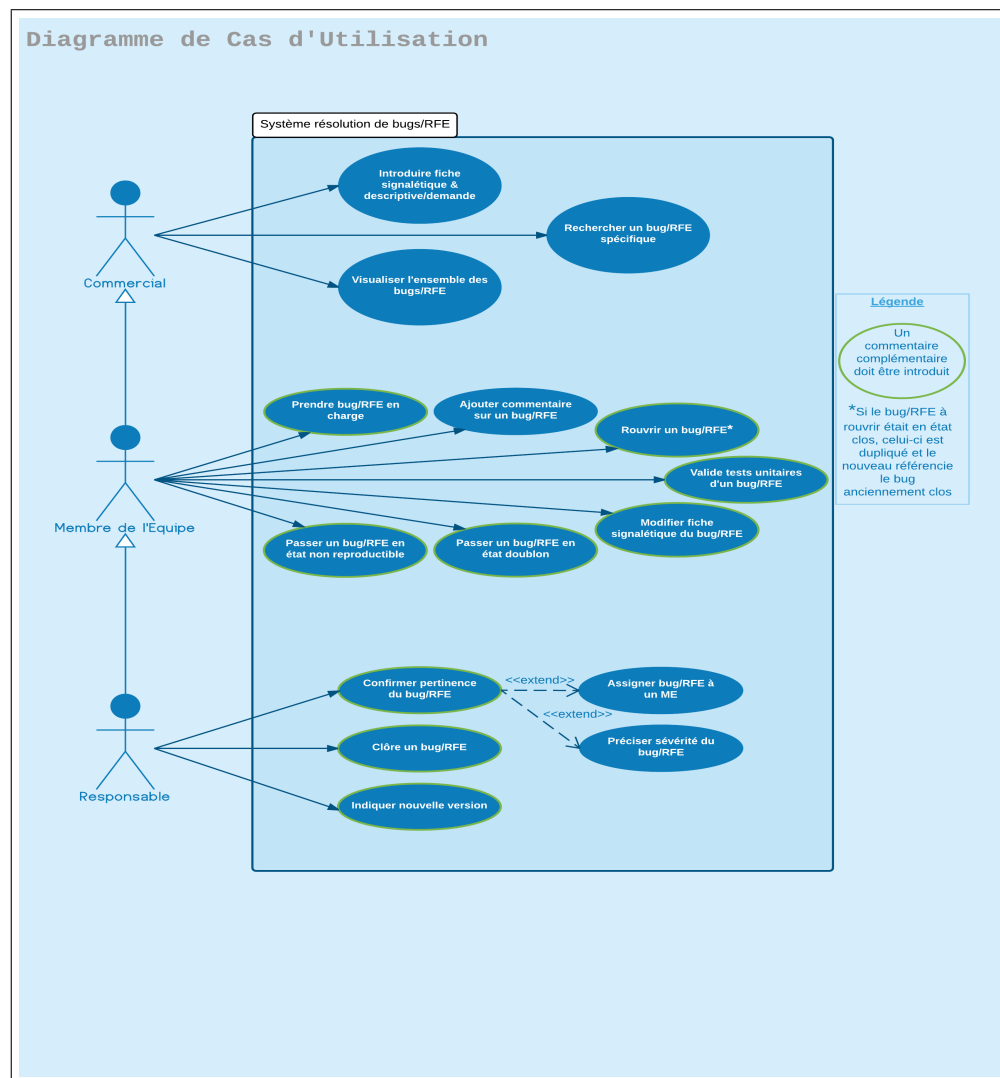


FIGURE 1 – Diagramme des cas d'utilisation

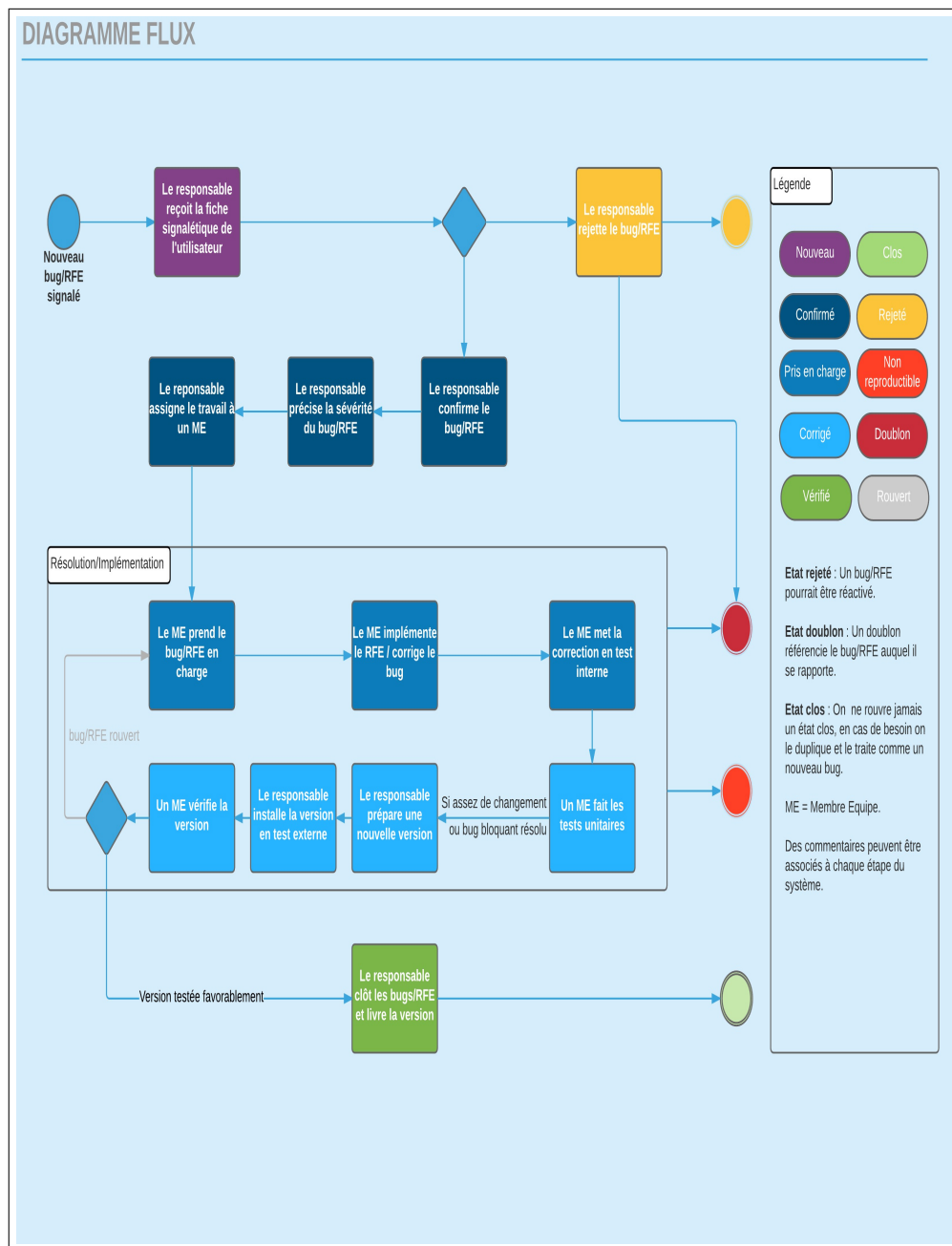


FIGURE 2 – Diagramme de flux de la gestion d'un bug



## 4.3 Description du travail des utilisateurs

### 4.3.1 L'intermédiaire

L'intermédiaire n'est pas un utilisateur actif de l'application, sa fonction est de faire le relais entre les clients et les bugs qu'ils rencontrent et le commercial qui les signale à l'équipe de développement. Pour cela, l'application prévoit une utilisation de suivi de bug pour que l'intermédiaire puisse informer un client de l'avancement de la résolution d'un bug en particulier. L'intermédiaire **n'est pas authentifié** au sein de l'application. N'importe quel utilisateur ayant accès à un bug concernant un intermédiaire peut en seul geste lui envoyer un e-mail généré automatiquement résumant l'état du bug actuel.

**NOTE : AJOUTER UNE IMAGE DU BOUTON FAIRE PART A L'INTERMEDIAIRE + UN EXEMPLE DE MAIL GENERE AUTOMATIQUEMENT**

### 4.3.2 Le commercial

Le commercial est un utilisateur **authentifié** de l'application (grâce à son identifiant unique et son mot de passe). Il est le point de relais entre l'intermédiaire et l'équipe de développement, c'est lui qui signale tous nouveaux bugs.

**Signalement de bug** Pour cela il suffit depuis l'écran principal d'appuyer sur le bouton « Ajouter un bug /RFE », l'application lui invitera ensuite à remplir un formulaire correspondant à la fiche descriptive du bug.

**Note :** placer ici la vue avec la fiche descriptive et signalétique d'un bug (en mettant en évidence le bouton ajouter bug) et en grisant la possibilité de changer la sévérité d'un bug

**Visualisation et recherche de bugs** Pour l'aider dans sa tâche, l'application lui permettra de visualiser l'ensemble des bugs et d'effectuer des recherches (par état, par logiciel, le concernant ou pas... **A VERIFIER/ PRECISER COHERENCE GRAPHIQUE IHM**).

**Note :** placer ici la vue principal en insistant sur la partie visualisation/recherche

**Espace personnel** Le commercial a également accès à son propre espace personnel depuis lequel il peut modifier ses identifiants de connexion à l'application. Depuis cet écran, il peut avoir accès à un certain nombre de statistiques le concernant. (**A VERIFIER/ PRECISER COHERENCE GRAPHIQUE IHM**)

### 4.3.3 Le membre d'équipe de développement

Le membre d'équipe tient un rôle central dans l'application, c'est lui qui gèrera la majorité des transitions d'état des bugs afin de pouvoir garder un détail sur l'évolution de ceux-ci. Il dispose également des mêmes fonctionnalités que le commercial. La fonction de recherches de bugs possède une option

supplémentaire par rapport au commercial qui lui permet de voir rapidement quels bugs lui ont été assignés par son responsable. **A VERIFIER/ PRECISER COHERENCE GRAPHIQUE IHM)**

**Insertion de commentaires** Pour l'aider dans le développement du bug et pour tenir informer le reste de l'équipe, le membre d'équipe peut laisser un commentaire grâce à une vue prévue à cet effet. Cette action est obligatoire lorsqu'il change l'état d'un bug (un champ est directement prévu dans le menu de transition d'état pour les commentaires obligatoire ainsi que dans la fenêtre commentaires pour les facultatifs). Il est à noter que chaque commentaire lié à un bug conserve l'état du bug durant lequel il a été écrit. La vue ci-dessous donne un exemple de l'historique des commentaires d'un bug. Un code couleur correspondant à l'état des bugs indique durant quel étape le commentaire a été écrit. **Note : placer ici la vue concernant les commentaires**

**Transition d'état** Le menu principal du bug comprend différents boutons pour marquer la transition d'état d'un bug. Le membre d'équipe peut ainsi choisir de marquer un bug comme « pris en charge » (s'il décide de prendre en charge son développement), « corrigé » (s'il considère que le bug est corrigé), « validé » (s'il a terminé de valider les tests unitaires en interne concernant le bug), « non-reproductible » (si le bug est considéré comme annulé), « rouvert » (si un bug en état « non-reproductible » est jugé reproductible ou qu'un bug n'a pas pu être validé après avoir été corrigé) ou en état « doublon » (si le bug a déjà été signalé). Cette dernière transition impliquera de devoir sélectionner le bug initial faisant double-emploi avec le bug courant. Il est à noter que ce n'est pas parce qu'un membre d'équipe qui a pris en charge un bug et l'a corrigé ne doit pas obligatoirement se soucier de réaliser les tests internes. Une fois qu'un bug a reçu l'état « corrigé », il peut être repris par un autre membre d'équipe. **Note : placer ici la vue concernant les transitions de bug (griser celui de clore pour responsable) + monter qu'on peut bel et bien référencer le bug initial pour l'état doublon**

**Modification d'une fiche** Un membre d'équipe peut modifier la fiche signalétique ou descriptive d'un bug à tout moment. **Note : placer ici la vue où l'on peut voir un bouton pour la modification d'une fiche**

#### 4.3.4 Le responsable d'équipe de développement

Le responsable d'équipe qui est également un utilisateur **authentifié** est doté des mêmes fonctions que le membre d'équipe. A cela près qu'il est doté d'outils supplémentaires liés aux préparations de nouvelles versions (pouvant regrouper plusieurs bugs) ainsi qu'à l'assignation de bugs à des membres d'équipes.

**Transitions d'états** Trois transitions d'états lui sont réservées « confirmé » et « assigné » et « clôt ». En effet le responsable peut dans un premier temps

confirmer un nouveau bug en précisant sa sévérité mais décider de l'assigner à un membre d'équipe à un autre moment. L'état « clôt » correspond à l'état d'un bug lorsque le responsable confirme la correction de celui-ci et qu'il s'apprête à préparer une nouvelle version du logiciel prenant en compte les modifications effectuées. **A VERIFIER/ PRECISER COHERENCE GRAPHIQUE IHM + DSD + DIAGRAMME UC + DIAGRAMME FLUX**) Note : placer ici la vue où l'on peut voir le champs sévérité de la fiche signalétique non grisé + la possibilité de sélectionner un membre d'équipe pour lui assigner un bug (attention différent prise en charge, il faut rajouter un nouvel état == à discuter)

**Préparation de nouvelles versions** Une fois qu'un ou plusieurs bugs dont l'état est « clôt » le responsable peut décider de préparer une nouvelle version du logiciel concerné. Pour l'aider dans sa tâche, un système de glisser-déposer lui permettra de reprendre les bugs repris dans la nouvelle version aisément. **Note :** placer ici la vue où l'on prépare une nouvelle version, avec le glisser-déposer

#### 4.4 Description des données

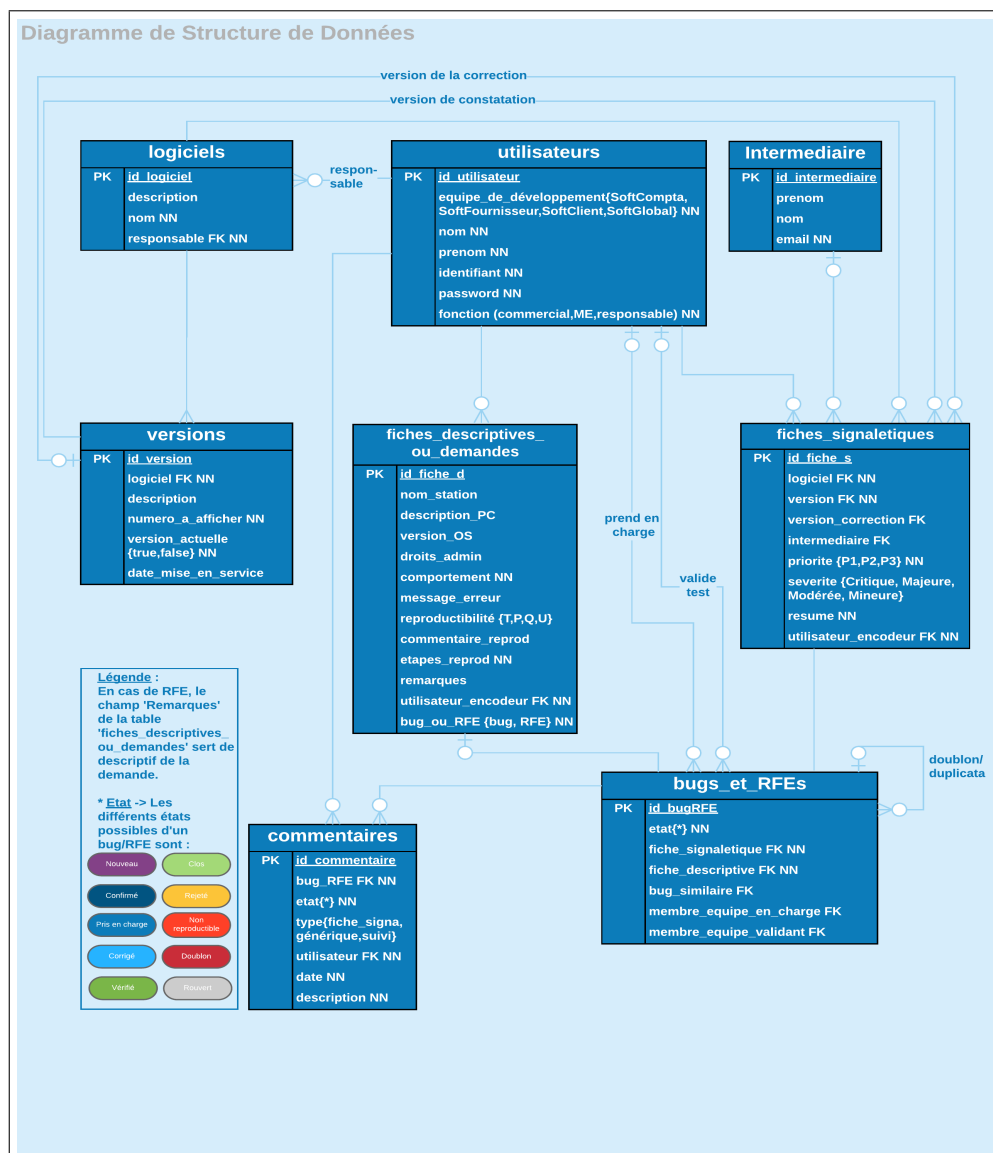


FIGURE 3 – Diagramme de structures des données

## 5 Glossaire

TO DO