

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

27 novembre 2016

Table des matières

1	Contexte	2
2	Analyse de l'existant	2
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
3	Objet du marché	5
3.0.1	Objectifs	5
3.0.2	Contraintes	5
4	Spécifications applicatives	6
4.1	L'application Bugtopus TM	6
4.2	Fonctionnalités	8
4.3	Le commercial	9
4.3.1	Visualiser l'ensemble des bugs/RFE	9
4.3.2	Introduire une fiche signalétique, une fiche descriptive ou un RFE	9
4.3.3	Rechercher un bug ou une RFE spécifique	9
4.4	Le membre d'équipe de développement	9
4.4.1	Prendre un bug ou une RFE en charge	9
4.4.2	Rouvrir un bug ou une RFE	9
4.4.3	Valider les tests unitaires pour un bug ou une RFE	9
4.4.4	Modifier une fiche signalétique ou descriptive	9
4.4.5	Passer un bug ou une RFE en état doublon	9
4.4.6	Passer un bug ou une RFE en état non-reproductible	9
4.5	Le responsable d'équipe de développement	9
4.5.1	Confirmer la pertinence du bug ou d'une RFE	9
4.5.2	Clôre un bug ou une RFE	9
4.5.3	Indiquer une nouvelle version	9
4.6	Persistance des données	11
5	Spécifications de réalisations	12
6	Glossaire	12

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 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
- Clos
- Rejeté

- Non-reproductible
- Doublet
- 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.¹

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

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

4 Spécifications applicatives

4.1 L'application **Bugtopus**TM

4.2 Fonctionnalités

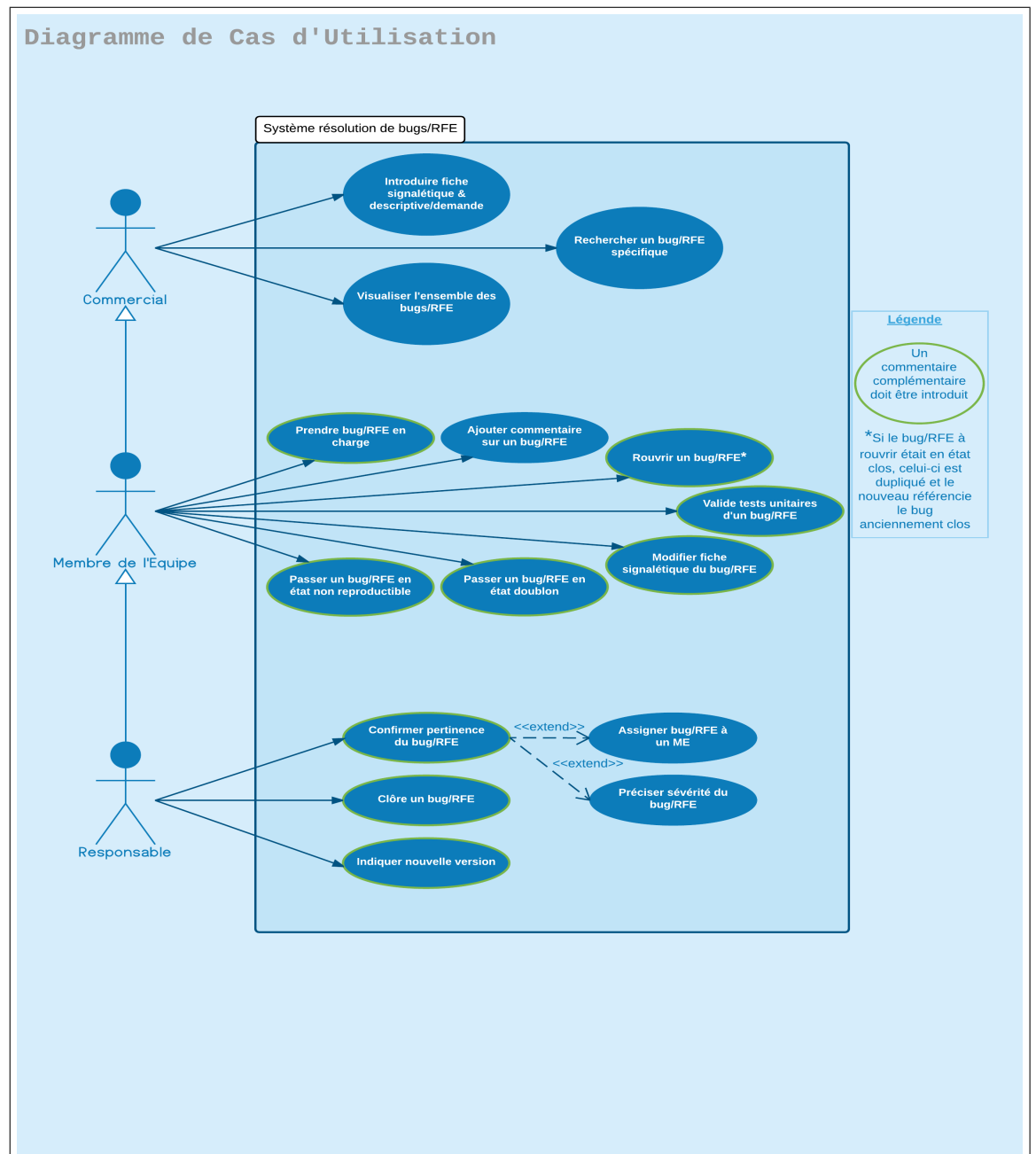


FIGURE 1 – Diagramme des cas d'utilisation

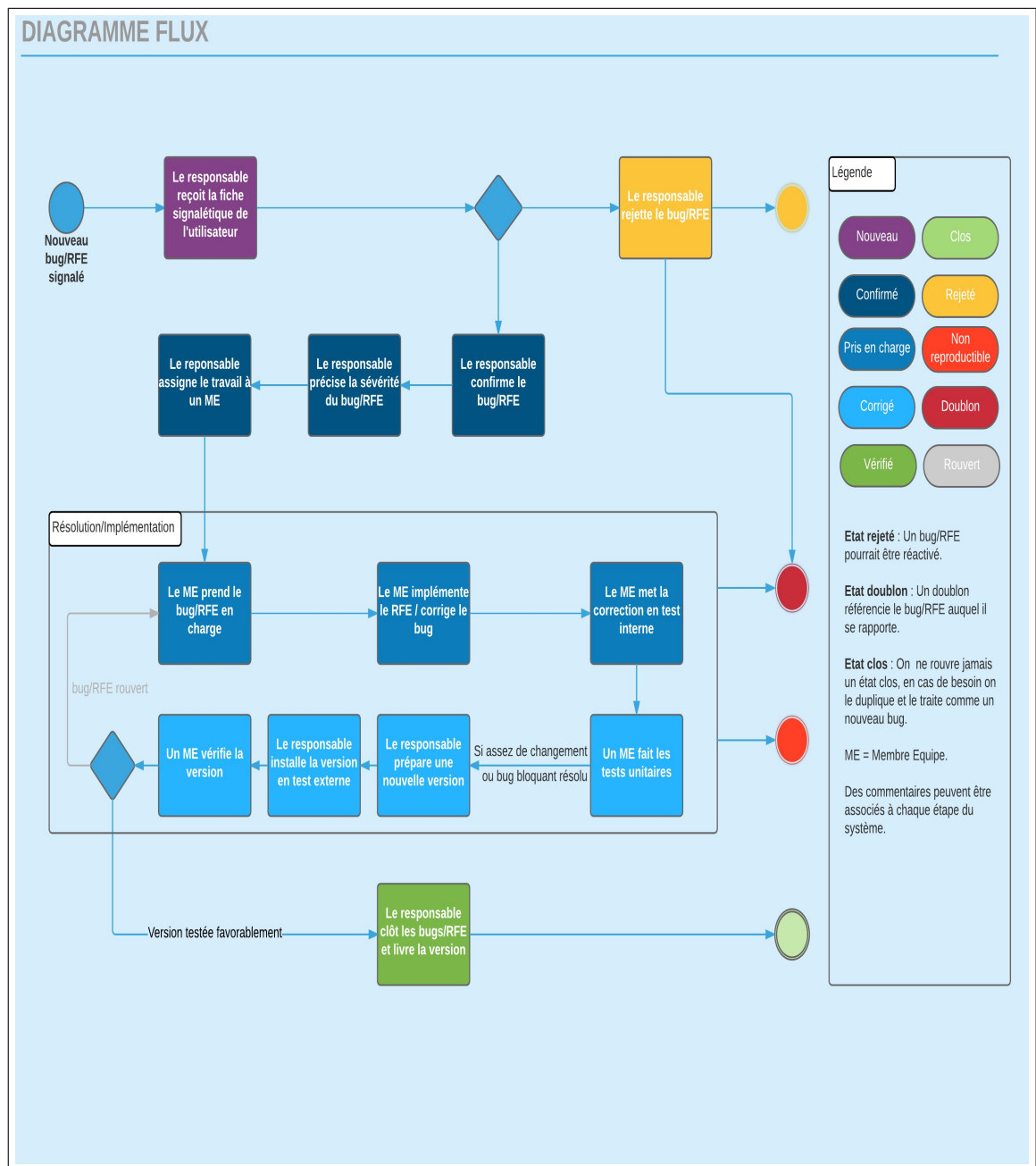


FIGURE 2 – Diagramme de flux de la gestion d'un bug

4.3 Le commercial

4.3.1 Visualiser l'ensemble des bugs/RFE

4.3.2 Introduire une fiche signalétique, une fiche descriptive ou un RFE

4.3.3 Rechercher un bug ou une RFE spécifique

4.4 Le membre d'équipe de développement

4.4.1 Prendre un bug ou une RFE en charge

4.4.2 Rouvrir un bug ou une RFE

4.4.3 Valider les tests unitaires pour un bug ou une RFE

4.4.4 Modifier une fiche signalétique ou descriptive

4.4.5 Passer un bug ou une RFE en état doublon

4.4.6 Passer un bug ou une RFE en état non-reproductible

4.5 Le responsable d'équipe de développement

4.5.1 Confirmer la pertinence du bug ou d'une RFE

4.5.2 Clôre un bug ou une RFE

4.5.3 Indiquer une nouvelle version

4.6 Persistance des données

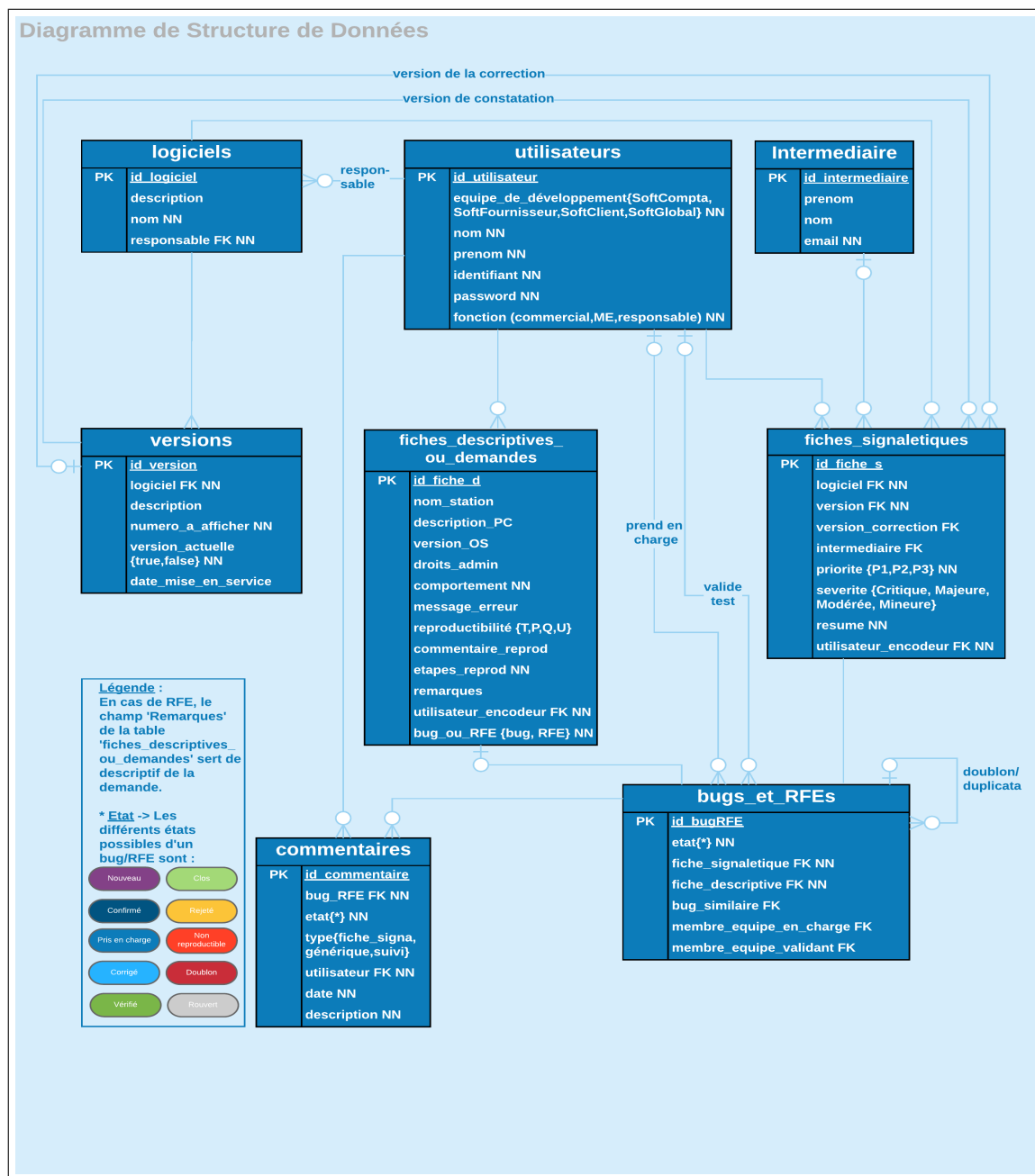


FIGURE 3 – Diagramme de structures des données

5 **Spécifications de réalisations**

6 **Glossaire**