

Table des matières

1. Introduction	2
2. Présentation.....	2
3. Installation.....	2
3.1. Perl.....	2
3.2. Moteur de base de données.....	3
3.2.1. MySQL	3
3.3. Serveur Web	3
3.4. Bugzilla.....	3
3.5. Modules Perl	4
3.6. Bugzilla utilisant apache.....	5
4. Complémentaires.....	6
4.1. Installation du paquetage linguistique français.....	6
4.2. Installation de la documentation.....	6
4.3. Personnalisation des modèles	6
5. Administration et configuration	6
6. Intégration de Bugzilla avec d'autres outils	8

1. Introduction

Un système de suivi de problèmes (ou système de suivi de bogues) est un outil qui permet d'aider les développeurs et les utilisateurs à améliorer la qualité d'un logiciel. Les utilisateurs soumettent les défauts de fonctionnement rencontrés dans le logiciel. Les développeurs sont alors toujours au fait des problèmes rencontrés.

Il existe des nombreux outils open sources de gestion des anomalies à savoir *Bugzilla*, *WebIssues*, *eTraxis*, *Anomalia*, *Trac*, *Mantis*, *Bug Tracker*, *Bugnet*, *The Bug Genie*, *Issue Tracker*, *Eventum*. Mais Bugzilla se distingue par le faite qu'il permet un suivi plus efficace des bogues et il est très mature et riche en termes de fonctionnalités offertes.

2. Présentation

Bugzilla est une application écrite en Perl et elle constitue le standard en termes de système de suivi des bogues (Bugzilla est utilisé par beaucoup d'entreprises pour suivre le développement en interne de leurs logiciels à savoir Mozilla, Open Office, la NASA, Facebook et des centaines d'autres sociétés populaires).

3. Installation

Le logiciel serveur Bugzilla est habituellement installé sur GNU/Linux ou Solaris. Lui faire fonctionner sur Windows est possible mais ça nécessite l'ajustement de plusieurs modules, ce qui rend l'installation plus difficile que sur Unix. Pour cette raison, nous vous avons opté pour une installation sur un système compatible Unix (nous avons fait le choix d'Ubuntu).

Les grandes lignes, l'installation se déroule ainsi :

- Installation de Perl (5.8.0 ou supérieur pour les plateformes non-Windows)
- Installation d'un moteur de base de données (Mysql, Postgres, etc.)
- Installation d'un serveur Web (généralement un serveur apache)
- Installation de Bugzilla
- Installation des modules Perl
- Installation d'un agent de transfert de courrier (Sendmail 8.7 ou supérieur par exemple)

Ci-dessous, on détaille la procédure d'installation de chacun de ces éléments requis.

Noua avons commencé par installer un système compatible Unix sur notre machine. L'installation de Bugzilla diffère en quelques points d'un système à un autre, mais en gros l'idée consiste à suivre les mêmes étapes qu'on venait d'indiquer. Une version plus récentes et surtout mise à jour facilite énormément le processus de la mise en place de cette solution de gestion des anomalies.

Nous avons procédé par la suite à une mise à jour de notre système. La mise à jour des packages se fait grâce à la commande suivante : *Sudo apt-get update*

3.1. Perl

Test de la version Perl installée : *perl -v*

Si le système d'exploitation utilisé ne fournit pas de paquetages officiels, on doit visiter le site <http://www.perl.com>. Bien que Bugzilla fonctionne avec Perl 5.8.0, c'est une bonne idée que d'utiliser la dernière version stable.

3.2. Moteur de base de données

À partir de Bugzilla 2.20, le support de serveur de bases de données MySQL et PostgreSQL est inclus.

Pour faire fonctionner Bugzilla, nous avons fait le choix de Mysql comme système de gestion de notre base de données.

3.2.1. MySQL

Test de la version installée : `mysql -V`

Si Mysql n'existe pas, nous devons lancer cette commande : `sudo apt-get install mysql-server`.

Si MySQL n'existait pas encore et que le système d'exploitation utilisé ne fournit pas de paquetages officiels, on doit aller sur le site <http://www.mysql.com> pour pouvoir les télécharger. Nous sommes besoin de MySQL version 4.1.2 ou supérieur pour faire fonctionner correctement Bugzilla.

3.3. Serveur Web

Nous avons la liberté de choix ici, pratiquement n'importe quel serveur capable d'exécuter des scripts CGI fonctionnera. Cependant, il est fortement recommandé d'utiliser un serveur Apache (version 1.3.x ou 2.x).

Test de la version installée : Consultez la page d'accueil par défaut sur `http://<notre-machine>/` ou bien taper la commande : `apache -v`. Dans certaines systèmes tel que le notre, le serveur est nommé apache2 au lieu d'apache et donc il faut utiliser la commande utilisé est `apache2 -v`.

Si apache ou apache2 n'existait pas sur la machine utilisée, il faut taper dans le terminal cette commande : `apt-get install apache2`

Si on n'a pas encore apache et que le système d'exploitation qu'on travaille avec ne fournit pas de paquetages officiels, on doit aller sur le site <http://httpd.apache.org/> et procéder à l'installation de ce serveur.

Tapons ensuite la commande :

`Sudo /etc/init.d/apache2 s` pour voir les différentes fonctionnalités à utiliser pour manipuler votre serveur.

Par exemple :

`Sudo /etc/init.d/apache2 status` permet de savoir l'état du serveur

`Sudo /etc/init.d/apache2 start` permet le démarrage d'apache2

`Sudo /etc/init.d/apache2 stop` permet d'arrêter le serveur.

Une fois le serveur installé et démarré, l'adresse `http://<notre-machine>/` doit pointer vers une page ou est écrit en gros et gras « It works ».

3.4. Bugzilla

Téléchargeons une archive de Bugzilla depuis le site de cet outil et plaçons-le dans un répertoire approprié accessible par l'utilisateur par défaut du serveur Web (« www » dans notre cas).

Bugzilla peut être placé ailleurs mais à condition d'ajouter un lien symbolique à partir de l'espace web.

N'oublions pas de rendre le répertoire où est placé Bugzilla, accessible en écriture pour l'utilisateur de notre serveur Web. Ceci est une étape temporaire jusqu'à l'exécution du script `checksetup.pl` qui verrouille l'installation.

La version téléchargée étant sous le format « .tar.gz », doit être désarchivée dans le répertoire adéquat en utilisant la commande suivante :

`Sudo tar -xvfz /home/bouabid/bugzilla-3.2.2.fr.tar.gz -C /var/www/`

`/home/bouabid/bugzilla-3.2.2.fr.tar.gz` : C'est le chemin vers la version téléchargée
`/var/www/` : le chemin vers l'emplacement où va être stockée notre version de Bugzilla à utiliser.

3.5. Modules Perl

Le processus d'installation de Bugzilla est basé sur un script appelé *checksetup.pl*.

Commençons par vérifier si nous avons les versions appropriées de tous les modules Perl requis.

Nous devons demeurer « *root* » jusqu'à la fin de l'installation (chaque commande est précédée par le mot « *sudo* » dans le cas où le système est Ubuntu tel que le notre).

Pour vérifier l'existence des modules requis, exécutons cette commande :

`checksetup.pl --check-modules`

Dans le cas où aucun module existe, l'exécution de cette commande finit par afficher que les commandes à installer sont les suivantes :

- `/usr/bin/perl install-module.pl CGI`
- `/usr/bin/perl install-module.pl Template`
- `/usr/bin/perl install-module.pl Email::Send`
- `/usr/bin/perl install-module.pl Email::MIME`
- `/usr/bin/perl install-module.pl Email::MIME::Modifier`
- `DBD::mysql` (dans le cas d'une utilisation de mysql)
- `Chart: /usr/bin/perl install-module.pl Chart::Base`
- `Template-GD: /usr/bin/perl install-module.pl Template::Plugin::GD::Image`
- `MIME-tools: /usr/bin/perl install-module.pl MIME::Parser`
- `PatchReader: /usr/bin/perl install-module.pl PatchReader`
- `Perl-ldap: /usr/bin/perl install-module.pl Net::LDAP`
- `Authen-SASL: /usr/bin/perl install-module.pl Authen::SASL`
- `RadiusPerl: /usr/bin/perl install-module.pl Authen::Radius`
- `SOAP-Lite: /usr/bin/perl install-module.pl SOAP::Lite`
- `HTML-Scrubber: /usr/bin/perl install-module.pl HTML::Scrubber`
- `Email-MIME-Attachment-Stripper: /usr/bin/perl install-module.pl`
- `Email::MIME::Attachment::Stripper`
- `Email-Reply: /usr/bin/perl install-module.pl Email::Reply`

Pour une installation automatique de tous ces modules, il est conseillé d'utiliser cette commande :

`/usr/bin/perl install-module.pl -all`

Il est probable que cette ligne de commande échoue l'installation de certains modules. Dans un cas pareil, nous devons procéder à une installation manuelle qui consiste à télécharger le module en question du site approprié et l'installer ou avoir recours à cette méthode qui est considérée la plus facile et qui consiste en l'ouverture du synaptic (le gestionnaire de paquets) et le lancement d'une recherche sur le module à installer. Puis il ne reste qu'à cocher les paquets intéressants et à cliquer sur "Appliquer".

Note : Au moment de l'installation du module `DBD::mysql`, nous avons été invités à saisir le mot de passe pour pouvoir accéder à la base de données qui sera créée.

L'installation de tous les modules étant achevée, la ré-exécution du script *checksetup.pl* confirmera à nouveau que tous les modules sont présents et remarquera la modification du fichier *localconfig*.

Attribuons à ce fichier le droit d'exécution en utilisant la commande :

Chmod u+x localconfig puis l'éditons en mode « root » moyennant la commande qui suit :

Sudo vi localconfig afin de d'ajuster les paramètres :

- \$webservergroup à vide (\$webservergroup='') pour une installation de test.
Notons bien que le fait de mettre cette variable à vide est vraiment non sécurisé car n'importe quel utilisateur ayant un accès local à notre machine peut manipuler à sa volonté nos fichiers et toucher ainsi à notre configuration. Pour cela il faut définir un group auquel appartenait notre serveur afin d'y limiter et sécuriser l'accès.
- \$db_user à « root » (\$db_user = root)
- \$db_pass au mot de passe que nous venons de définir pour accéder à la base. (\$db_pass = 'mot de passe défini')

Ce script compile ensuite les modèles de l'interface utilisateur, se connecte à la base de données en utilisant l'utilisateur « bugs » créé par défaut et le mot de passe qu'il nous demande de définir et crée enfin la base de données « bugs » et les tables à l'intérieur.

Après cela, il demande des détails sur le compte administrateur. Bugzilla peut avoir plusieurs Administrateurs. Nous pouvons en créer plus tard. Mais il en a besoin d'un pour démarrer. Saisissons l'adresse électronique d'un administrateur, son nom complet, et un mot de passe approprié pour Bugzilla. *checksetup.pl* se terminera alors et c'est possible de la relancer à tous les moments si on le souhaite.

3.6. Bugzilla utilisant apache

Il existe deux options pour exécuter Bugzilla sur Apache : *mod_cgi* (par défaut) et *mod_perl*. Nous avons eu recours à la première manière malgré la puissance de *mod_perl* qui assure une meilleure rapidité et un plus grand contrôle du serveur.

L'utilisation du *mod_cgi* consiste à éditer le fichier *httpd.conf* localisé normalement dans */etc/apache2* et d'ajouter ce bout de code :

```
<Directory /var/www>  
AddHandler cgi-script .cgi  
Options +Indexes +ExecCGI  
DirectoryIndex index.cgi  
AllowOverride Limit  
</Directory>
```

Ces instructions autorisent apache à exécuter les fichiers « .cgi » se trouvant dans le répertoire Bugzilla et indiquent au serveur de chercher un fichier appelé *index.cgi* si quelqu'un saisit seulement le nom du répertoire dans la barre d'adresse du navigateur.

4. Complémentaires

4.1. Installation du paquetage linguistique français

Pour obtenir une version française, en plus de la version anglaise par défaut de Bugzilla, il faut télécharger le pack correspondant depuis le site officiel de cet outil, le décompresser dans le répertoire correspondant à l'installation de Bugzilla. On doit avoir à présent le répertoire `bugzilla/template/fr` avec la nouvelle version du paquetage linguistique français. Il ne faut pas oublier d'exécuter le script `checksetup.pl` avant de se connecter.

4.2. Installation de la documentation

Il est préférable que l'installation de Bugzilla soit accompagnée d'un manuel d'utilisation qui pourra aider certains utilisateurs. Pour ce faire, il suffit de télécharger les paquetages de documentations qu'on souhaite installer, décompresser chacun dans le répertoire de Bugzilla, et d'associer au paramètre `docs_urlbase` le chemin vers la documentation installée. Localement et dans notre cas par exemple, `docs_urlbase` est égale à <http://localhost/bugzilla/bugzilla-3.2.3/docs/%lang%/html/>. « %lang% » sera remplacé par la langue préférée de l'utilisateur.

4.3. Personnalisation des modèles

Il est possible de configurer l'aspect et la convivialité de Bugzilla sans avoir à éditer des fichiers Perl en s'appuyant sur l'une de ces deux méthodes :

- La première méthode consiste à éditer directement le contenu de répertoire « *default* » présent sous `template/en/default`. C'est à priori la meilleure méthode pour de petites modifications.
- La deuxième méthode consiste à copier les modèles à modifier dans une arborescence de répertoires miroir sous `template/en/custom`. Les modèles de cette arborescence prennent le pas sur tous les modèles de même nom et situés au même endroit dans le répertoire `default`.

5. Administration et configuration

Il ne reste maintenant qu'accéder à la page `http://<notre-machine>/`, se loguer en tant qu'administrateur et procéder à l'administration de Bugzilla qui dépendra énormément des nos besoins d'utilisations. Nous listons dans la suite les différentes fonctionnalités nécessaires pour la configuration de Bugzilla en vue de l'estimation du temps nécessaire en nombre de jours pour mettre en place cette solution et l'adopter à nos besoins.

- Définition des groupes d'utilisateurs
- Restriction des groupes (restreindre la visibilité des bogues à un groupe d'utilisateurs, associer des projets à un groupe particulier)
- Administration des utilisateurs (création, recherche, modifications des comptes utilisateurs, etc.) et affectation de ces derniers aux groupes.
- Mécanisme d'authentification à utiliser (base de données ou LDAP, gestion des cookies, définition des expressions régulières pour la validité des adresses électroniques)

Nombre de jours de travail estimé pour la gestion des utilisateurs et des groupes d'utilisateurs : **2 jours**

- Gestion des produits (projets), de leurs composants et de catégories de produits

- Gestion des versions, des jalons, et des étiquettes relatives à un produit. Les jalons sont les objectifs pour lesquels on projette de corriger un bogue alors que les étiquettes sont un moyen d'attacher un état spécifique à un bogue ou un fichier joint.

Nombre de jours de travail estimé pour la gestion des projets : **1 jour**

- Définition des champs d'une bogue (Ajouter des champs personnalisés)
- Définition des options par défaut des requêtes sur les bogues
- Recherche avancée des bogues
- Politique des modifications des bugs (politique sur le comportement par défaut des événements de modification de bogues).
- Gestion des étiquettes sur les bogues et les fichiers joints
- Définition des mots clés pour faciliter le traçage des bogues
- Configuration du workflow des états d'une bogue

Nombre de jours de travail nécessaire pour la définition d'un processus de gestion des bogues : **3 jours**

- Configuration de la notification par courrier électronique
- Les notifications programmées (événements qui s'exécutent quotidiennement)

Nombre de jours de travail estimé pour la configuration du service mail et l'envoi de notifications : **0.5 jour**

- Mise à jour de Bugzilla (définir les moyens d'obtenir la nouvelle version de Bugzilla : correctifs, téléchargement de l'archive ou par CVS)

Nombre de jours de travail estimé pour définir les méthodes de gestion de la mise à jour de Bugzilla : **1 journée**

- Sécuriser Bugzilla (gestion de la sécurité des serveurs web et de base de données, accès réseau, prévention contre l'injection des codes malicieux)

Nombre de jours de travail estimé pour sécuriser Bugzilla : **2 jours**

- Gestion des rapports (état actuel de la base) et des graphiques (état de la base en fonction du temps).

Nombre de jours de travail estimé pour la configuration du graphique des bogues : **0.5 jour**

- Configuration d'une base de données esclave (c'est une sorte de répllication de données qui permet d'autoriser la lecture des données dans une table qui est en cours de modification par un autre collaborateur)
- Définition d'un ensemble de bulletins de votes afin de mieux jauger les besoins des utilisateurs du produit en cours de développement.
- Personnalisation de quelques modules

Nombre de jours de travail estimé pour réaliser les précédentes tâches : **2 jours**

IL en résulte que le temps estimé pour la mise en place d'une telle solution est 12 jours de travail. Rajoutons à ceux-ci une journée pour refaire l'installation de l'environnement du travail et des serveurs en question, nous finissons par conclure qu'il nous faudra environ 15 jours de travail pour la réalisation de notre application de suivi de bogues en se basant sur Bugzilla.

6. Intégration de Bugzilla avec d'autres outils

Il est possible d'intégrer Bugzilla avec plusieurs autres outils de développement logiciels telqu'un serveur svn (Subversion) pour la gestion de versions lors de la mise en place d'un projet de développement informatique.