

# Systeme de gestion de tickets

ALBERT SHIH<sup>1</sup>

<sup>1</sup>Observatoire de Paris - Meudon

18 mars 2009

# RT, kesako ?

- Système de gestion de tickets, RT=Request Tracker.
- GPL.
- perl (mod\_perl), MySQL (postgresql, SQLite, Oracle).
- Installation
  - Nécessite  $\approx$  140 packages CPAN.
  - Serveur Apache 2.x.
- Migration complexe ( $X.Y.Z \rightarrow X.Y + 1.Z'$ ). Actuel : RT-3.8.2
- Configuration
  - Compréhension utile.
  - Certains choix définitifs.

# Gestion de tickets

- Gestion des demandes.
- Le suivi des demandes.
- Le suivi des réponses.
- Priorité dans les demandes.
- Archivage des demandes/réponses.
- Pour plusieurs personnes.
- Qui s'adapte aux conditions.
- Ce n'est pas un planificateur de tâche (Diagramme de Gantt).

# Tickets

- Objet de base dans RT.
- Identifié par numéro et nom de domaine.
- UN demandeur.
- UN seul intervenant (éventuellement *nobody*).
- Des observateurs.
- Des «admin»-observateurs.
- ≈ 20 champs de base (un seul obligatoire).
  - Priorité de départ, priorité final.
  - Date de début, date de fin.
  - Temps passés.
  - N'importe quelles champs.
  - Statut : *Nouveau, Ouvert, Stagnant, Rejeté, Effacé, Résolu*.
  - Des *commentaires* et des *réponses*.

## Utilisateurs/groupes

- Quelqu'un utilise RT :
  - Quelqu'un qui soumet/ouvre un ticket (demandeur).
  - Quelqu'un qui travail avec RT (intervenant).
- Identification par adresse mail/login.
- Création de ticket par mail  $\implies$  création «à la volé» des comptes.
- Impossibilité de créer un ticket via le web pour un demandeur.
- Compte  $\nRightarrow$  accès.
- Accès à RT  $\implies$  compte.
- $\approx$  50 types de droits différents pour chaque utilisateurs.

# File d'attente

- Seconde objet le plus important.
- Un ticket  $\in$  UNE file d'attente.
- 24 types de droits différents par file d'attente.
- Notion de
  - Priorité (début, fin).
  - Délais.
  - Observateurs.
  - Champs personnalisés.
  - etc. . .
  - Scripts.

# Scripts

- Nécessaire pour customiser votre RT.
- Programmable «à la main».
- $\approx$  10 scripts de bases.
- Permet de définir des «situation, actions».
- Utilisation : Envoie de mails à réception de tickets, Envoie de mail pour rajout de commentaires, résolution etc. . . .
- Utilisation de templates.
- Indispensables.
- Scripts de bases suffisants.

# Fonctionnement

- Création de ticket → mail ou web.
- Création par mail → `procmail+script perl`.
- Attention aux spam.
- Existe un outil CLI.
- Transactions/Échanges via mail.
- Administration via interface web.
- Existe un module pour passer des commandes via mail.



# RTFM

- RTFM  $\neq$  Read That F\*\*\* Manual.
- RTFM = RT FAQ Manager.
- Permet de créer des FAQ depuis RT.
- Module supplémentaire de RT.

## Vie d'un ticket

- Alice envoie d'un mail à `support@votrelabo`  $\implies$  Création d'un ticket, (*nobody*, Nouveau).
- RT renvoie le mail aux «observateurs» de la file d'attente, et un accusé de réception à Alice
- Un observateur Bob «prend» le ticket (via interface web). Les observateurs reçoivent un mail d'information. ( Bob, ouvert )
- Bob pose une question à Alice en répondant au mail de RT.
- RT renvoie le mail aux observateurs et à Alice.
- Alice répond au mail de RT.
- RT renvoie le mail aux observateurs.
- Bob envoie un commentaire à RT.
- RT renvoie le mail aux observateurs.

## Vie d'un ticket

- Charles «vole» le ticket (via le web), (Charles, ouvert).
- RT renvoie un mail d'information aux observateurs.
- Charles résout le problème.
- Charles répond au mail de RT informant la résolution du problème.
- RT renvoie le mail aux observateurs et à Alice.
- Charles ferme le ticket (via le web), (Charles, résolu).
- RT renvoie un mail de fermeture à Alice, et un mail d'information aux observateurs.

# Les opérations

- On peut «donner» un ticket à quelqu'un ou *nobody*.
- On peut changer un ticket de file d'attente.
- On peut passer un ticket de ouvert à nouveau.
- On peut attacher des documents via l'interface web, ou par mail.
- On peut rajouter/supprimer un/des observateurs pour un ticket.
- Notion de dépendance entre les tickets.
- Possibilité de «merger» des tickets.
- Possibilité de changer tout les champs d'un tickets.
- Impossibilité de supprimer des réponses/commentaires.

Utilisation par une personne

## Mode de fonctionnement

- Pas de problème de droits.
- Pas de besoin de nombreuses files d'attente.
- Utilisation de la fonction d'avancement automatique des priorités.
- Création à la main des tickets.
- Création de tickets systématiquement.
- Une tâche → un ticket.
- Priorité finale identique.
- Délais de réalisation différente suivant file.

Utilisation par une personne

## alors ? vs mutt+vi ?

- Avantages :
  - Boites aux lettres quasi-vides, mail → mail (communication).
  - Avancement automatique des priorités.
  - Plus «jamais» la question «*je commence par quoi*».
  - Plus d'oublis.
  - On traite les tâches, même ceux qu'on a pas envie.
  - Archivage des étapes.
  - Archivage du temps passés sur chaque problème.
  - Ticket valable «depuis».
  - Archivage de toutes les tâches réalisées.
  - Archivage des correspondances par tâches (sans avoir  $N \gg 1$  boites aux lettres)
  - Génération automatique du «fait.txt».
  - Ça flatte l'égo.

Utilisation par une personne

## alors ? vs mutt+vi ?

- Désavantages :
  - Nécessite une vraie discipline pour créer les tickets.
  - Impersonnel.
  - Uniquement en ligne (en particulier pour RTFM).
  - C'est déprimant.
  - Groupe :
    - Impossible d'utiliser les scripts.
    - Intégration manuelle des commentaires (par mail).
    - Nécessite un «vrai» agent de mail.

Utilisation par un groupe (PLM-TEAM)

# Mode de fonctionnement

- Droits simples.
- Utilisation des scripts.
- Utilisation de la fonction d'avancement automatique des priorités.
- Création des tickets par mail indirecte (nécessite un «vrai» agent de mail).
- Une tâche → un ticket.



Utilisation par un groupe (PLM-TEAM)

## alors ? vs mutt+vi ?

- Avantages :
  - Permet la gestion de l'attribution/prise des tâches.
  - Plus d'oublie.
  - Archivage des étapes.
  - Plus de problème «group-reply» vs «reply».
  - Archivage des correspondances par tâches.
  - Au courant des changements.
- Désavantages :
  - Problème «group-reply».
  - Uniquement en ligne.
  - Nécessite un vraie discipline.

# Divers

- Utilisation du mail.
- Réouverture d'un ticket à la correspondance.
- Par défaut pas d'envoi de mail à l'utilisateur.
- Possibilité de génération de mot de passe à la volée.
- Mailing-list assez active.
- Passage de version complexe.
- Mise en place des scrips long.
- Mise en place de file potentiellement long.
- Petits défauts
  - Ressource système.
  - Pas de split de ticket.
  - Pas de gestion de congès.

# Conclusion/Remarques

- Conclusions :
  - Entièrement satisfait.
  - Ne remplace pas `vi` pour les docs longs.
- Remarques
  - RT ne... résout pas les tickets.
  - RT n'impose rien (dépend de la configuration).
  - Existence de notion l'«approbation».
  - Culture ?

## Démo/Questions

