

# Méthodes et outils employés pour développer des logiciels libres

Étude basée sur le cas du  
projet Debian

Par Raphaël Hertzog  
Gérant de Freexian SARL  
Développeur Debian

# Présentation du projet Debian

- Objectif : créer un système d'exploitation entièrement libre
- Les contributeurs sont bénévoles et acceptent les valeurs défendues par les deux textes fondateurs du projet
  - le contrat social  
[http://www.debian.org/social\\_contract](http://www.debian.org/social_contract)
  - la définition d'un logiciel libre selon Debian

# Plan de la présentation

- Structure et organisation
  - Communautés structurées autour de différents moyens de communication
- Méthodes de travail
  - Transparence, ouverture à tous, contributions encouragées et facilités
- Outils employés
  - Pour la documentation, la diffusion, la programmation, etc.
- Transposition à une entreprise ?

# Structure et organisation

- La communauté est la première ressource d'un projet libre comme Debian
  - 1000 développeurs officiels, plusieurs milliers de contributeurs occasionnels
  - des dizaines de milliers d'utilisateurs (certainement beaucoup plus mais aucun chiffre officiel n'existe)
- Une communauté n'existe qu'au travers d'un outil de communication permettant aux membres d'échanger
- Si trop de monde => création de « sous-communautés spécialisées »

# Outils de communication

- Listes de diffusion : <http://lists.debian.org>
  - plus de 160 listes; 1,2 millions de mails envoyés chaque jour (36000 messages différents par mois);
- Discussion en temps réel (IRC) : [irc.debian.org](http://irc.debian.org)
  - #debian-devel: 200 personnes connectées à toute heure de la journée
- Site web : <http://www.debian.org>
  - Ainsi que de nombreux sites « officiels »
- Blogs : <http://planet.debian.org>

# Hiérarchie très légère

- Souvent un dictateur bénévole
  - Initiateur du projet qui a l'autorité ultime sur l'évolution du logiciel (ex: Linus Torvalds sur le noyau Linux)
- De temps en temps, une structure un peu plus formalisée
  - Debian: leader élu annuellement par les développeurs (selon les règles de la constitution Debian)
- Toujours une méritocratie : les galons se gagnent en faisant quelque chose d'utile pour le projet (« celui qui fait, décide »).

# Méthodologies de développement

- Modèle « bazar » (vs « cathédrale »).
  - <http://www.linux-france.org/article/these/cathedrale-bazar/>
  - Cathédrale: modèle traditionnel, organisé et centralisé (spécifications, conception, développement, tests)
  - Bazar: modèle du logiciel libre (mosaïque des contributions individuelles)
- Ouverture
  - Peu de préjugés car communication relativement impersonnelle
  - Disponibilité des codes sources et de tout l'historique
  - Large base d'utilisateurs testeurs des versions en développement

# Méthodologies (suite)

- **Transparence**
  - Le processus de développement est public ainsi que les discussions ayant menées à un choix donné dans le passé
  - La liste des choses à faire et des bogues est publique
- **Contributions encouragées et facilitées**
  - Intégration rapide des améliorations => satisfaction du travail utile pour le contributeur
  - Chaque utilisateur est un contributeur potentiel



# Cas de Debian

- Système de suivi de bogue public : <http://bugs.debian.org>
- Distributions en développement (« testing / unstable ») mises à jour quotidiennement
  - sources et binaires disponibles sur de nombreux miroirs (serveurs de téléchargement)
  - employées par de nombreux utilisateurs avancés
- Processus de parrainage pour accueillir de nouveaux développeurs
- Archives en ligne de toutes les listes de diffusion : <http://lists.debian.org>

# Outils employés

- Pour développer :
  - gestion de la configuration (cvs, svn, arch, ...)
  - système de suivi de bogues
- Pour documenter :
  - outil d'édition collaboratif (wiki)
  - rédaction document balisé + chaîne de conversion
- Pour assurer le suivi :
  - système de « traceur » (souvent lié au système de suivi de bogues)

# Gestion de la configuration

- Gestion de codes sources
  - Permet à plusieurs personnes de travailler en parallèle et de synchroniser leur travail (gestion des conflits, possibilité de fusion avancée)
  - Permet de maintenir plusieurs versions en parallèle (une stable, une version de développement, etc.)
  - Traçabilité de tous les changements (génération de diffs, consultation des messages de logs associés, etc.)

# Gestion de la configuration (suite)

- Outils libres : SVN (Subversion), Arch, CVS (Concurrent Version System), ...
- Interfaces web
  - Navigation dans l'archive avec consultation des versions historiques ; possibilité de voir les différences entre plusieurs versions et de retrouver qui est responsable d'un bout de code donné
  - <http://cvs.debian.org>
  - <http://svn.debian.org>

# Systeme de suivi de bogues

- Répertorier et classer les problèmes connus pour mieux les corriger
- Systeme de priorité et de gravité
  - Gravité : importance de l'impact du problème sur les utilisateurs
  - Priorité : classement indépendant pour trier selon les priorités des développeurs
- Catégorisation par sous-système
- Répartition entre plusieurs développeurs
- Affectation d'un bogue à une version future (planification de correction)

# Systeme de suivi de bogues (suite)

- Debian utilise « debbugs »
  - totalement piloté par courrier électronique
  - interface web pour la consultation uniquement <http://bugs.debian.org>
  - adapté à la structure de Debian (organisation autour de nombreux paquets logiciels)
- Autre logiciel populaire :
  - bugzilla : <http://www.bugzilla.org/>
  - tout s'effectue dans une interface web

# Outils pour la documentation

- La documentation est une tâche souvent négligée, rarement faite par les développeurs eux-mêmes
- Comme pour les codes sources, il faut des outils permettant à plusieurs personnes de collaborer efficacement
  - Soit la documentation est intégrée dans les sources sous forme de document XML (docbook)
  - Soit la documentation est intégrée dans le site web et un « wiki » permet alors à tout le monde de l'étendre et de la corriger facilement

# Documentation avec langage balisé

- Un langage balisé comme le « Docbook-XML » permet de rédiger un document avec un simple éditeur de texte et de contribuer à plusieurs via l'outil habituel de gestion de la configuration
- Une chaîne de traitement XSLT permet ensuite de générer la documentation dans différents formats (HTML pour le web, PDF pour l'impression, etc.).
- Avantage : documentation intégrée dans les sources
- Inconvénient : docbook est un langage nécessitant un période d'apprentissage



# Documentation dans un wiki

- Un wiki est un site web qui peut-être directement édité et complété par ses visiteurs au moyen de simples formulaires web
  - Ex: <http://wiki.debian.org>
- La syntaxe d'un wiki est très légère et facile à appréhender
- Avantage : extrême facilité pour contribuer
- Inconvénients : risques liés aux contributions anonymes (suppression de contenu, introduction d'erreurs subtiles, ...), la documentation est « externe »

# Notion de forge

- Une forge est un « hébergeur de projets » qui offre à chaque projet l'ensemble des services traditionnellement employés par un logiciel libre
  - Gestion de la configuration, gestion des bogues, des requêtes de support
  - Site web, wiki, listes de diffusion, etc.
- Gforge est le logiciel le plus connu
  - <http://gforge.org>
- Trac est plus limité mais très bien intégré
  - <http://projects.edgewall.com/trac/>

# Transposition à une entreprise

- Les outils du logiciel libre correspondent parfaitement à une situation de « télé-travail » ou les différents collaborateurs sont distants
- Mais qui peut le plus peut le moins, les mêmes outils peuvent être employés dans une entreprise traditionnelle
  - Freexian peut vous installer et vous administrer votre « forge interne »
  - Ces outils vous permettent en outre de proposer le télé-travail à certains de vos salariés (et de vous y préparer pour plus tard)

# Transposition à une entreprise (suite)

- Les méthodes ne fonctionnent par contre que pour des vrais logiciels libres (disposant d'une réelle communauté d'utilisateurs)
- C'est pourquoi, il peut-être intéressant de « libérer » les logiciels spécifiques qui ne vous donnent pas d'avantage concurrentiel mais qui représentent une charge pour vous !
  - Possibilité d'économiser en réduisant le coût de maintenance grâce à une mutualisation avec d'autres utilisateurs dans la même situation

# Conclusion / Questions

- Le monde de l'entreprise doit apprendre à profiter du logiciel libre :
  - en l'employant pour ses qualités intrinsèques
  - en comprenant ses règles spécifiques pour bénéficier au maximum d'une mutualisation intelligente
  - en s'inspirant de son mode de développement pour gérer les projets qui s'y prêtent
- Questions ?

# Liens / Contact

- Le projet Debian
  - <http://www.debian.org>
- Freexian SARL
  - SSII spécialisée dans le logiciel libre et Debian GNU/Linux
  - <http://www.freexian.com>
  - Contact: Raphaël Hertzog <[raphael@freexian.com](mailto:raphael@freexian.com)>
  - Tél: 06 88 21 35 47