



Les logiciels libres : caractéristiques, utilisation dans le développement et place dans les Systèmes d'Information

Jean-Luc Archimbaud

► To cite this version:

Jean-Luc Archimbaud. Les logiciels libres : caractéristiques, utilisation dans le développement et place dans les Systèmes d'Information. Engineering school. Angers, 2010, pp.82. <cel-00553071>

HAL Id: cel-00553071

<https://cel.archives-ouvertes.fr/cel-00553071>

Submitted on 6 Jan 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cours (6-8 h) pour étudiants en informatique niveau Master 2

Les logiciels libres : caractéristiques, utilisation dans le développement et place dans les Systèmes d'Information

Jean-Luc Archimbaud CNRS/UREC

Directeur du projet PLUME

Jean-Luc.Archimbaud@urec.cnrs.fr

<http://www.projet-plume.org>

Janvier 2010

Ce document est sous licence Creative Commons Paternité-Pas d'Utilisation
Commerciale : <http://creativecommons.org/licenses/by-nc/2.0/fr/>

Plan du cours

- ◆ 2 mots sur PLUME (projet dans le libre)
- ◆ Législation logiciels, définitions et origine
- ◆ Les licences libres
- ◆ Développement des logiciels libres : comment, qui, pourquoi ?
- ◆ Points forts et faibles des logiciels libres
- ◆ Où les trouver ?
- ◆ Développer (avec) des logiciels libres
- ◆ Métiers-sociétés autour des logiciels libres
- ◆ Place du libre dans les entreprises et les administrations
- ◆ Principaux types de logiciels libres utilisés
- ◆ En prenant un peu de recul...

PLUME

- ◆ Promouvoir les Logiciels Utiles, Maîtrisés, Economiques dans Ens Sup – Recherche
- ◆ Serveur (demo plus tard)
 - ❖ Fiches descriptives de logiciels utilisés, ressources
--> mutualisation des compétences
 - ❖ FAQ, transparents, Ecole... sur développement (libre)
--> communauté logiciel (libre) Ens Sup – Rech
 - ❖ Fiches Développements Ens Sup Recherche
--> promotion développements internes
- ◆ Utilisation / cours
 - ❖ Serveur : source d'informations logiciels et utilisations
 - ❖ Exemple projet : avec logiciels libres et dans son esprit

Législation

- ◆ **Logiciel**
 - ❖ Code source (lisible, modifiable) ou code exécutable (binaire, illisible)
- ◆ **CPI : Code Propriété Intellectuelle (1992)**
 - ❖ Propriété industrielle (inventions, marques...)
 - ❖ Propriété littéraire et artistique (droits d'auteurs...)
- ◆ **Logiciel = oeuvre de l'esprit**
 - ❖ CPI : code + dossier d'analyse + documentation...
 - ❖ Couvert par le droit d'auteur
 - ❖ Droits moraux + droits patrimoniaux

Législation

- ❖ Droits moraux
 - « Protéger « la personnalité » de l'auteur au travers de son œuvre et respecter celle-ci »
 - --> Indication nom(s) concepteur(s) du logiciel dans la diffusion
- ❖ Droits patrimoniaux
 - Droits d'exploitations – copyright
 - Diffuser, faire commerce...
- ❖ Logiciel : qui a les droits patrimoniaux ?
 - Développeur salarié pendant travail -> Employeur
 - Développeur pendant temps libre -> Développeur
 - Sur commande (société tierce) --> Dépend du contrat
- ❖ Domaine public : pas de propriété
 - Législation logiciel : au bout de 70 ans
 - (Ancien) abus de langage : domaine public = logiciel libre

Législation

- ❖ Loi : droit d'utilisation d'un logiciel inclut
 - Copier pour sauvegarde
 - Sauf licence restrictive
 - ✓ Corriger les erreurs
 - ✓ Analyser le fonctionnement
 - ✓ Reproduire et traduire code pour inter-opérabilité
- ❖ Art. L. 335-2 du CPI
 - Toute personne utilisant, copiant, modifiant ou diffusant le logiciel sans autorisation explicite du détenteurs des droits patrimoniaux est coupable de contrefaçon et passible de trois ans d'emprisonnement et de 300 000 € d'amende
 - Autorisation pour logiciel --> licence
- ❖ Attention : la législation dépend du pays et évolue
 - Souvent suivant la puissance des lobbys
 - Bcq (trop) de débats : anti logiciels libres, cabinets avocats...

Définitions

◆ Licence

- ❖ Contrat (de louage) entre détenteurs des droits patrimoniaux du logiciel - utilisateur
- ❖ Fixe les modalités d'utilisation (usage)
- ❖ Peut concéder tout ou partie des droits patrimoniaux

◆ Licence libre : 4 libertés sur le logiciel

- Définition de la FSF, Free Software Fondation (85)
- ❖ L'exécuter pour tous les usages
- ❖ L'étudier, l'adapter --> accès au code source
- ❖ Le redistribuer (copier) gratuitement ou non
- ❖ L'améliorer et publier les améliorations --> source

Définitions

- ◆ **Logiciel libre (LL) : logiciel avec licence libre**
 - ❖ Imposé : code source diffusé
 - ❖ Téléchargeable n'implique pas Libre
 - ✓ Peut être freeware, shareware...
 - ❖ Logiciel libre n'implique pas gratuit !!!
 - ❖ Contraire : logiciel propriétaire
- ◆ **Autre nom : Open Source**
 - ❖ Free signifie gratuit et libre --> Open source préférable ?
 - ❖ Déf Open Source Initiative (1998) : similaire au départ
 - ❖ Mais maintenant orientations commerciales
- ◆ **FLOSS : Free/Libre Open Source Software**
- ◆ **Attention aux 'abus de langage' de commerciaux**
- ◆ **Remarque : fork : 1 --> 2 versions indépendantes**

Origine

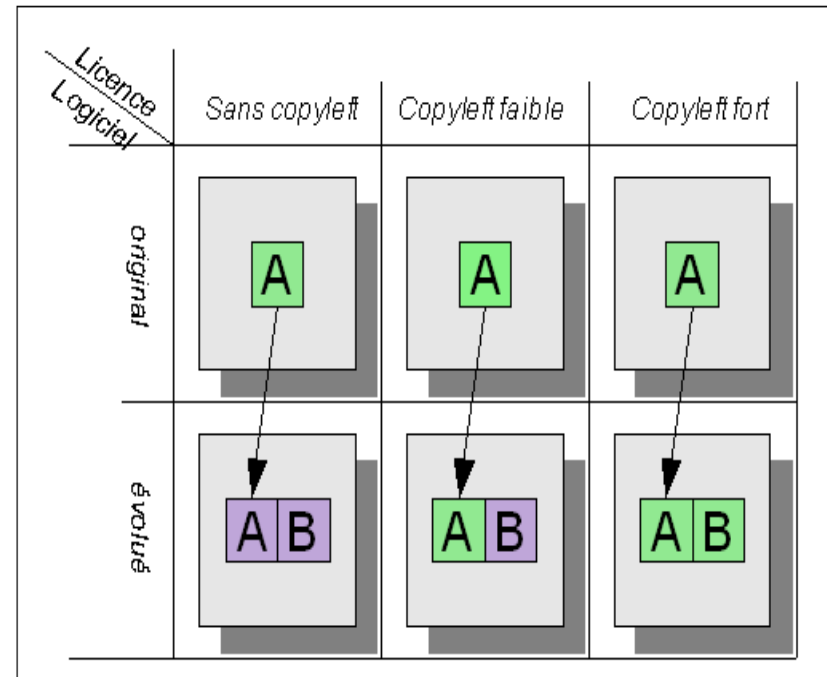
- ◆ < 1975 : le logiciel n'a pas de valeur
 - Achète du matériel
- ◆ 1975 – 1980 : valeur faible
 - Constructeurs : groupes d'échanges entre développeurs
 - Mais arrivée : micro-informatique, Microsoft...
- ◆ > 80 : payant (cher) et boîte noire
 - Pbs : bug, pas d'adaptation possible, pas d'échange...
- ◆ 83 : début : projet GNU - R. Stallman (MIT)
 - OS compatible Unix + logiciels avec sources partageables
 - ✓ --> 92 : Linux (qu'on devrait appeler GNU-Linux)
 - 85 : création FSF et 89 : licence GPL

Les licences libres

- ◆ Plus de 100 différentes (mais 6 majeures)
- ◆ Origine géographique
 - US : GPL (69 % des logiciels libres), LGPL (10 %), BSD (9 %), MIT (2 %), Apache (1 %), Mozilla (1 %)
 - FR : CeCILL (CEA, CNRS, INRIA) : 3 versions
 - EU : EUPL en plusieurs langues
- ◆ Une licence : document de 10-15 pages
 - Souvent uniquement en anglais pour licences US
- ◆ Validité devant un tribunal en France ?
 - Pas de procès, jurisprudence / licence libre

Les licences libres

- ◆ 3 types / impact sur ré-utilisation du code
- ◆ Copyleft fort
- ◆ Licence initiale s'impose sur tout
- ◆ Licence contaminante
- ◆ Copyleft faible
- ◆ Licence initiale reste, les ajouts peuvent avoir d'autres licences
- ◆ Sans Copyleft
- ◆ Licence initiale ne s'impose pas
- ◆ Les dérivés peuvent avoir n'importe quelle licence



Les licences libres

❖ Exemples

- Copyleft fort : GPL, CeCILL V2
- Copyleft faible : LGPL, Mozilla, CeCILL-C
- Sans copyleft : Apache MIT, CeCILL-B
- Important pour les développeurs qui veulent diffuser-vendre
 - ✓ On ne peut pas utiliser n'importe quelle brique / diffusion qu'on veut en faire

❖ Licences Creative Commons

- Pour les créations : video, musique, article, documentation...
- 3 paramètres
 - ✓ Paternité : citer le nom auteur
 - ✓ Utilisation commerciale ou non
 - ✓ Modification ou non

❖ Utilisation d'un logiciel sans licence ?

Dév de LL : comment ?

- ❖ Deux origines pour le logiciel
 - Associatif (fondation, consortium) : historique et gros logiciel
 - Entreprise : de plus en plus
- ❖ Méthode de développement : peu de théorisée
 - Pas de MOA-MOE-cahier des charges fonctionnel...
 - Bon sens, pragmatisme - Économie (moyens minimums) – Réactivité
 - Mais se rapproche du développement agile
- ❖ Outils – Moyens (cf transparents suivants)
 - Equipe de développement = contributeurs organisés
 - Un outil de base : Internet
 - Un outil de développement collaboratif : forge
 - Diffusion des versions en temps réel
 - Des organisations : communautés...

Origine associatif / entreprise

◆ Associatif

- ❖ Développeurs = équipe virtuelle
- ❖ 'Bénévoles', non rémunérés (pour ce travail), dispersés
- ❖ Différent d'une équipe de développement d'une société

◆ Entreprise

- ❖ Développeurs : au départ viennent de la société
- ❖ Si le logiciel se diffuse largement → même organisation qu'un développement de type associatif

◆ Mais ce n'est pas si tranché

- ❖ Des (grosses) sociétés soutiennent des projets associatifs
 - IBM → Eclipse, Google → Mozilla, SUN → OpenOffice

Equipe de développement

- ◆ Un développeur seul
 - ❖ Les utilisateurs lui remonte des bugs, il corrige seul
- ◆ N développeurs - 1 'responsable'
 - ❖ Des développeurs (contributeurs) proposent du nouveau code pour améliorer l'existant (des contributions)
 - ❖ Le responsable les intègre ou pas, gère les contributeurs, définit les évolutions et les versions
- ◆ Une arborescence hiérarchique de contributeurs et de responsables
 - ❖ Linux : +2000 contrib -> 300 mainteneurs de pilote -> mainteneurs de ss-système -> mainteneurs principaux

Un outil de base Internet

- ◆ Internet 'utilisable' : 90 – Licence GPL : 89
 - ❖ Partout, très peu cher
 - ❖ Pas de logiciels libres à grande échelle sans Internet
- ◆ Des contributeurs partout dans le monde
 - ❖ Facile de réunir les 3 développeurs US, EU, JP intéressés par un sujet pointu
- ◆ Correctifs, versions...
 - ❖ Diffusion instantanée et gratuite
- ◆ Des outils de communication
 - ❖ Messagerie électronique, listes de diffusion, wikis...
 - ❖ Travail coopératif

Forge : service

- ◆ Site Web sur lequel on peut
 - ❖ Créer un espace de travail (un projet)
 - ✓ Avec un administrateur et plusieurs contributeurs
 - ❖ Gérer du code, le diffuser (faire connaître)
 - ✓ Gestion des sources, des versions, des bugs, des tickets
 - ✓ Gestion des documents (Wiki), forums, listes, sondages, news
 - ❖ Par projet (l'administrateur du projet choisit)
 - ✓ Carte des logiciels, outils de recherche, demandes aide...
 - ❖ Avec des règles
 - ✓ Réservé à une communauté, pour certaines licences...
 - ❖ Avec des objectifs différents :
 - ✓ Développement collaboratif
 - ✓ Diffusion uniquement
 - ✓ ...

Forge : exemples de service

- ◆ Sourceforge : US : <http://sourceforge.net/>
 - ❖ Le plus connu : plus de 180 000 projets
 - ❖ Logiciels sous licence libre uniquement
 - ❖ Gratuit – ouvert à tous
 - ❖ Très peu de gouvernance
 - ✓ On ne détruit rien : 90 % des projets sont morts
- ◆ SourceSup : FR : <http://sourcesup.cru.fr/>
 - ❖ Réservé Ens Sup et Recherche : forge de communauté
- ◆ INRIA : FR : <http://gforge.inria.fr/>
 - ❖ Réservé INRIA (Inst. Rec. Info.) : forge d'entreprise
 - ❖ Permet de 'contrôler' et conseiller les développeurs

Forge : exemples de logiciels

- ◆ Au départ logiciel libre : SourceForge
 - ❖ --> 2 forks : 2 versions qui vivent leur vie différemment
 - ❖ Un reste libre mais évolue (ci-dessous)
 - ❖ L'autre : propriétaire --> SourceForge Enterprise Edition
- ◆ Le fork libre a engendré
 - ❖ Gforge : GPL, gforge.org
 - ❖ Codendi
 - ❖ Savane
 - ❖ Fusionforge

Diffusion

◆ Par internet

- ❖ 'Gratuit', immédiat (en temps réel)
- ❖ Avant : bande, disquette

◆ Version stable

◆ Version(s) beta

- ❖ Plusieurs par an, très rapidement diffusées
- ❖ 1. Remontée immédiate des bugs
- ❖ 2. Corrections rapides
- ❖ 3. Nouvelle version
- ❖ Versions diffusées tôt (avec de nombreux bugs) : ce sont les utilisateurs (beta) qui font les tests

Organisations

- ◆ **Fondation (assoc 1901), consortium....**
 - ❖ Communauté développeurs mais aussi utilisateurs...
 - ❖ Quelques permanents : gestion-marketing-juriste,
 - ❖ Soutiens divers : membres, sponsors...
 - ❖ Membres (individus/sociétés)
 - Différents 'pouvoirs' : gold, silver...
 - Univ-recherche, gouv, sociétés utilisatrices, éditeurs de logiciels, intégrateurs...
- ◆ **Services : forge, gouvernance, image**
- ◆ **Réussite projet logiciel libre : personnalité 'charismatique' et bonne organisation**

Exemple d'organisation : OW2

- ❖ Open Source Middleware – Java : <http://www.ow2.org/>
 - ✓ INRIA-BULL-FT ---> structure indépendante
- ❖ Consortium : membres avec cotisation
 - ✓ 14 Strategic (sociétés Alcatel...) : contribution € et h/mois
 - ✓ 45 Corporate (univ-labos, gouv Min Int, entreprises)
 - ✓ 980 Individual
- ❖ 3 activités
 - ✓ Projets : open source traditionnels
 - ✓ Initiatives : lien fournisseurs de technologie-utilisateurs
 - ✓ Les chapitres locaux : présence de proximité, besoins de localisation, du code...
- ❖ Forge (100 projets), listes, réunions, salons...
- ❖ Board : strategic members
- ❖ Des permanents : organisation, admin forge, marketing

Développeurs de LL : qui ?

◆ Particuliers

- ❖ Passionnés de la programmation : gourou-hacker-geek
- ❖ Etudiants (concours google...)

◆ Employés

- ❖ Sous cape --> officiellement
- ❖ Chercheurs-enseignants (CNRS, universités)
 - Projet PLUME
- ❖ Certaines administrations
 - ADULLACT :le contribuable ne doit payer qu'une fois
- ❖ Entreprises : de plus en plus
 - Services de R&D, Sociétés de services en informatique, Editeurs informatiques...
 - ---> L'entreprise ne vend plus la licence mais du service (cf transparents métiers sociétés après)

Développeur LL : motivations ?

- ◆ Professionnelles : répondre aux besoins
 - ❖ De son métier
 - ❖ De son entreprise, d'un client (sociétés services informatiques)
 - ❖ --> nouveau logiciel ou évolution (contribution)
- ◆ Personnelles
 - ❖ Pour développer ---> créer
 - ❖ Apprendre (techniquement)
 - ❖ Echanger, être ensemble : dans une communauté
 - ❖ Ethique : donner au monde
 - ❖ Anti Windows, SAP, ORACLE...
 - ❖ Valorisation personnelle en dehors de son travail

Points forts des LL

- ◆ Liberté / éditeurs (logiciels propriétaires)
 - ❖ Pas obligé d'acheter tous les produits Microsoft parce que vous vous êtes basé sur Active Directory, d'acheter une licence pour chaque nouveau poste utilisateur, de migrer à Vista --> Win 7 en payant, de changer de version de SAP (plusieurs mois de travail) en payant, d'attendre la prochaine version qui intègre une modif vitale pour vous...
 - ❖ Critiques financières / logiciels propriétaires
 - Le coût de la licence n'est plus justifiée --> racket
 - Les utilisateurs/sociétés/DSI veulent payer le prix réel

Points forts des LL

- ◆ Majorité des licences : utilisation gratuite
 - ❖ Peu de coût « de base » : économique
 - ❖ Possibilité d'essayer immédiatement
 - ❖ 'Achat' immédiat
 - Important dans l'administration, grosses sociétés où les procédures d'achat sont longues
 - ❖ --> Utilisation très rapide, souvent gratuite
- ◆ Le code est lisible : peut être vérifié
 - ❖ Comprendre exactement ce qu'il fait
 - ❖ Sécurité : pas de boîte noire, de cheval de Troie caché
 - ❖ Détecter les bugs --> les remonter, les corriger
 - ❖ --> Compréhension, sécurité, fiabilité

Points forts des LL

- ◆ **Le code est lisible : peut être modifié**
 - ❖ Correction des erreurs soi-même
 - ❖ Modifications locales
 - ❖ Modifications intégrées dans la version de référence
 - Le produit inclut de nouvelles fonctionnalités qui correspondent à des vrais besoins
 - ❖ --> Adaptation, améliorations rapides / vrais besoins
- ◆ **Le code est lisible : transparence**
 - ❖ On peut comparer 2 produits, leur codage, leur conception modulaire...
 - ❖ Les meilleurs techniquement restent !
 - ❖ --> Qualité

Points forts des LL

- ◆ Diffusion rapide des versions
 - ❖ Time to market très court
 - ❖ En avance / produits commerciaux
 - ❖ Le produit « colle au marché »
 - ❖ --> Innovation !
- ◆ Ethique
 - ❖ Le logiciel est un bien public
 - ❖ Le logiciel doit être partagé sans contrainte
 - ❖ Mouvement FSF - Richard Stallman

Points forts des LL

- ◆ Comparaison avec la chaîne de production d'un logiciel propriétaire
 - ❖ Etude de besoins (service marketing)
 - ❖ Etude du marché (service marketing)
 - ❖ Spécifications fonctionnelles (service marketing)
 - ❖ Spécifications détaillées (service développement)
 - ❖ Développement (service développement)
 - ❖ Tests – validation (service test-validation)
 - ❖ Sortie d'une version
 - ❖ Vente de cette version (service commercial)
 - Qui doit être complète et avec peu de bugs
 - ❖ Un premier client utilise le logiciel 'en vrai'
 - ❖ --> une ou plusieurs années

Points forts des LL

- ◆ Ré-utilisation du code
 - ❖ Pour faire d'autres développements
 - ❖ Gratuit
 - ❖ Assure une pérennité du code d'origine
 - ❖ Effet 'standard'
 - ❖ --> Création de nouveaux logiciels (compatibles)
 - ❖ Fork possible si l'évolution ne va pas dans le sens désiré par les utilisateurs
 - ❖ --> Garantie / évolution (pas le cas de SAP, ORACLE...)

Points faibles - risques

- ◆ Remet en question l'industrie du logiciel
 - ❖ Ses modèles économiques et ses méthodes de développement
- ◆ Pérennité
 - ❖ Si le contributeur principal disparaît, se marie, change de job, n'est plus intéressé...
- ◆ Moins de motivation des contributeurs
 - ❖ Si non reconnus, si trop encadrés, si utilisation commerciale de leur travail...
- ◆ Il faut que les gens mangent
 - ❖ Tous les acteurs du libre doivent gagner leur vie --> modèles économiques variés

Points faibles - risques

- ◆ **Eparpillement de certains efforts**
 - ❖ Distributions Linux trop nombreuses avec des communautés trop concurrentes
- ◆ **Créer un 'monde nouveau', libre (Free soft)**
--> **modèle économique (Open source)**
 - ❖ Changement de dimension et d'objectif
 - ❖ Ne pas perdre les bases saines
- ◆ **Logiciels --> services – cloud computing**
 - ❖ Services google : le navigateur est le seul logiciel dont j'ai besoin !!!
 - ❖ Les entreprises n'achètent que des services : calcul, stockage, mail... : plus besoin de logiciel sur les postes

Où trouver des logiciels libres ?

- ◆ Bouche à oreille - Google
- ◆ Framasoft www.framasoft.org
 - ✓ Descriptifs, FR, généraliste, Windows surtout
- ◆ Wikipedia fr.wikipedia.org (portail logiciel libre)
 - ✓ Descriptifs, encyclopédie libre : lisse, pas de comparaison
- ◆ PLUME www.projet-plume.org : faire une démonstration
 - ✓ Descriptifs, FR, professionnel, Ens Sup – Recherche
- ◆ Developpez.com
 - ✓ Pour Développeurs
- ◆ FSF/UNESCO directory.fsf.org
 - ✓ Descriptifs
- ◆ Sourceforge sourceforge.net et Apache apache.org
 - ✓ Forge, généraliste ou licence apache
- ◆ Distributions Linux

Développer avec du libre

- ◆ **Comment choisir les composants libres**
 - Fonctionnalités évidemment
 - Compétences qu'on a : C++, JAVA, PHP... ?
 - Ouverture-standard
 - ✓ Formats, interfaces
 - Pérennité
 - ✓ Qui développe ? Comment ?
 - ✓ Qui utilise (quantité et 'qualité') ?
 - Licence : copyleft ou non
- ◆ **Quelle méthode de développement ?**
 - Exemple pragmatique : PLUME
 - Exemple théorisé : eXtreme Programming -développement agile

Exemple Dév PLUME

- ◆ **Choix et configuration de briques libres**
 - ❖ Pas vraiment du développement logiciel
- ◆ **Maquette : 6 mois**
 - ❖ Avec un existant SPIP – fiches=PDF
 - ❖ ---> Est-ce faisable, ça répond à des besoins ?
- ◆ **Rédaction CdC fonctionnel (en 2 jours)**
 - ❖ Utiliser du libre, min de développement, V1 rapide
 - ❖ Prévoit une architecture avec 3/4 composants
- ◆ **Choix briques : 3 mois (maquette continue)**
 - ❖ Etude des produits libres existants --> 5 produits retenus
 - ❖ Tests des 5 --> CMS Drupal (un seul composant) avec 30 modules

Exemple Dév PLUME

- ◆ Rédaction nouveau CdC : 3 jours
 - Reprend le 1er CdC et on trie / fonctionnalités de Drupal
 - ✓ Facile --> on fait
 - ✓ Difficile et pas besoin fondamental --> version suivante
 - ✓ Difficile et fondamental --> repense, contourne le besoin (processus...)
 - Décrit 2 phases précisément et un lot d'idées pour les versions suivantes
- ◆ Développement V1 : 4 mois
 - Plutôt installation et configuration (workflow...)
- ◆ Transfert maquette --> Drupal : 2 mois
 - Avec rédaction de pages, menus, mots-clés...
- ◆ Recette ---> documentation

Exemple Dév PLUME

- ◆ Ouverture du serveur (nov 07)
- ◆ Développement V2 : sur 9 mois
 - Classement des fonctionnalités (/ besoin et facilité)
 - Progressivement avec machine de développement
 - Intégration régulière des nouvelles fonctionnalités sur se serveur de production
- ◆ Recette ---> doc
- ◆ Dév V3 (identité visuelle, site en) : 6 mois
- ◆ Recette → doc
- ◆ Contacts permanents entre 'MOA' et 'MOE'
 - La MOA met la main à la configuration

Exemple Dév PLUME

- ❖ C'était la bonne méthode
 - Efficace et très rapide !
- ❖ On savait ce que l'on voulait
 - et on l'a écrit mais très simplement --> CdC
- ❖ **On a eu des surprises / CdC V1**
 - Drupal totalement inconnu pour nous, 3 briques --> une
- ❖ Les problèmes actuels viennent
 - Des développements faits (qd MAJ de modules Drupal)
 - De processus trop compliqués avec du dév (relecture)
 - --> Faire simple avec minimum de développement
 - Stabilité de la plate-forme : blocage : conf ? Robots ? ... ?
- ❖ **Et passage Drupal V5 à V6 : 8 mois ETP**
 - Beaucoup de configurations à refaire

eXtrem Programming

- ◆ **Contexte économique actuel**
 - ❖ Economie des moyens humains, financiers...
 - ❖ Efficacité / besoins – date mise à disposition
 - ❖ Réactivité : gestion du changement continue
- ◆ **Méthode de développement avec cycle en V mal adaptée**
 - ❖ Besoins -> spécif -> conception générale -> détaillée
-> codage -> tests unitaires -> intégration -> validation -> recette et enfin livraison
 - ❖ Modèle projet itératif PDCA (Plan, Do, Check, Ack) reconnu
- ◆ **--> XP – développement agile : 2001**
 - ❖ Un processus de développement, un état d'esprit et des valeurs, un ensemble de bonnes pratiques

XP : principes

- ◆ Le client travaille avec les développeurs
- ◆ Le développement est itératif
 - ❖ Les itérations sont courtes (2 semaines en moyenne)
- ◆ Les développeurs sont polyvalents et inter-opérables (compétences et connaissances)
 - ❖ Travail en binôme
- ◆ Chaque composant est vérifié par une campagne de tests automatisés
 - ❖ Programmation pilotée par les tests
- ◆ Chaque itération est validée par le client

XP : cycle de développement

- ◆ **Scenario : expression usage élémentaire**
 - ❖ Informel, rédigé en langue naturelle
 - ❖ Doit pouvoir être réalisé dans une itération
 - ❖ Implémentation n'est pas définitive
 - ✓ refactoring ultérieur : amélioration (simplification...) du code
- ◆ **Développement**
 - ❖ Expression initiale des besoins (1 mois)
 - ❖ N itérations (1ère longue, les suivantes : 2 semaines)
 - ✓ Choix de scénarii à réaliser
 - ✓ Ecriture code, écriture de tests (--> automatisés)
 - ✓ Intégration avec tests précédents
 - ✓ Livraison

XP : 5 valeurs

- ❖ La communication (entre les personnes)
 - Evite la plupart des problèmes
 - --> Travail avec client, en binôme, réunions de code...
- ❖ La simplicité (des solutions)
 - Meilleur moyen d'arriver à un résultat – Evolution facile
 - --> Faire simple, Ne pas anticiper les extensions futures
- ❖ Le feedback (avoir des retours réguliers)
 - ---> Test unitaires, fonctionnels, versions régulières (retours client)
- ❖ Le courage
 - De changer d'architecture, de jeter du code --> faire mieux, plus simple
- ❖ Le respect
 - Mutuel des membres, du travail

XP : 13 pratiques

- ◆ Un représentation du client sur site de développement
- ◆ Planning établi par client et équipe de développement
- ◆ Intégration continue (à chaque itération)
- ◆ Petites livraisons fréquentes
- ◆ Rythme soutenable (pour les développeurs)
- ◆ Tests fonctionnels à chaque itération
- ◆ Tests unitaires / programme
- ◆ Conception simple – doc minimale
- ◆ Utilisation de métaphores pour expliquer
- ◆ Refactoring régulier
- ◆ Appropriation collective du code
- ◆ Convention de nommage
- ◆ Programmation en binôme

XP : bilan

◆ Points négatifs

- ❖ Dogmatique : il faut appliquer tous les principes
- ❖ Difficile avec un prestataire (MOA et MOE)
 - Quel contrat ? Quel appel d'offre ?
 - Mal accepté par certains prestataires de services
- ❖ Importance primordiale du représentant du client

◆ Points positifs

- ❖ Agile (léger, adaptatif), 1ère version rapide
- ❖ Bien adapté au développement libre
 - Ex : Code_Aster (EDF) : analyse des structures et thermo-mécanique
 - 1,5 Millions de lignes <http://www.projet-plume.org/fr/fiche/code-aster>
- ❖ On peut reprendre des principes 'de bon sens'
 - cf PLUME

Développer du libre

- ◆ Les bonnes pratiques de développement (doc...) sont à appliquer
- ◆ Ne pas ré-inventer la roue
 - ❖ Bien étudier ce qui existe déjà (nouveau dév utile ?) et ce qui peut servir (dans le futur dév)
 - ❖ Ne pas être isolé : suivre-participer à une communauté
- ◆ Avoir une stratégie (prendre du recul)
 - ❖ Faire un CdC avec des priorités
 - ❖ Avoir une feuille de route : plusieurs phases
 - ❖ Diffuser rapidement une version --> V1 très simple
 - ❖ Etre agile : évoluer selon les retours...

Développer du libre

- ◆ Suivre-respecter les standards
- ◆ Choix licence
 - ❖ Quels sont les licences des briques utilisées ?
 - ❖ Quels sont les partenaires --> droits patrimoniaux
 - ❖ Que veut-on faire ?
- ◆ A moyen terme il faut décider
 - ❖ Où sera déposé le code diffusé ?
 - Sourceforge..., forge locale, Web local ?
 - ❖ Gestion des bugs ?
 - ❖ Gestion des contributeurs ?
- ◆ Indiquer Copyright et licence dans le code

Dév du libre : dans le code

- ◆ **En-tête pour tous les fichiers**
 - ❖ Nom du fichier, nom du logiciel
 - ❖ Copyright (ou ©, ou Droits patrimoniaux) + année(s) + nom de la personne morale ou physique
 - ❖ Auteur(s)
 - ❖ Licence
 - ❖ Utiles : date de création, une adresse de contact
- ◆ **Ajouter un fichier de licence**
 - ❖ Au package des fichiers source avec le texte complet ou une URL
- ◆ **Dans la doc, sur le site Web, indiquer :**
 - ❖ La licence
 - ❖ Les "briques logicielles" utilisées et leurs licences
 - ❖ Les contacts pour bugs et pour contribuer

Outils gestion de versions

◆ Buts

- ❖ Historique des modifications
- ❖ Travailler à plusieurs (verrous, gestion des conflits)
- ❖ Modifications en parallèle (branches)
- ❖ Sécurité (intégrité, disponibilité, confidentialité)

◆ Sur code source

- ❖ Mais aussi possible pour doc, fichiers de config..

◆ A utiliser, même si on travaille seul

◆ L'outil peut être intégré dans une forge

Outils gestion versions : jargon

- ◆ Patch
 - $V2 - V1 = \text{Patch}$ et $V1 + \text{Patch} = V2$
- ◆ Dépôt (repository)
 - Répertoire partagé par tous (historique des modifs)
- ◆ Révision
 - Chaque fichier a un numéro de révision unique
- ◆ Commit : soumission d'ajout-modification
- ◆ Update : mise à jour d'une version
- ◆ Branches : arborescence de versions du projet
 - Branche pour ancienne version sur laquelle on corrige encore des bugs, développer 2 idées en parallèle, gérer une version personnelle d'un logiciel, fusionner après une divergence

Outils gestion de versions

3 types

- ◆ Local : dans un système de fichiers local
 - ❖ Pas de réseau
 - ❖ SCCS, RCS
- ◆ Client/Serveur (ou centralisé)
 - ❖ Un serveur centralise le dépôt, accessible à distance
 - ❖ CVS --> **Subversion**
- ◆ Distribué
 - ❖ Multiples copies du dépôt, branches locales
 - ❖ bitkeeper, monotone, arch, darcs, mercurial, **git**, bazaar

Métiers-sociétés autour du LL

Ecosystème

Comment faire du business avec du 'gratuit' ?

- ◆ Société de services informatiques
- ◆ Hébergeur
- ◆ Vendeur de matériels
- ◆ Editeur de logiciels propriétaires
- ◆ Editeur de logiciels libres
- ◆ Editeur de distributions
- ◆ Société de formation
- ◆ Métier juridique
- ◆ Agence de communication

Sociétés de services (info) / LL

- ◆ SSLL (Sociétés de Services en Logiciels Libres) --> Toutes les sociétés de services
- ◆ Définition : le service informatique c'est vendre du temps et de l'expertise
 - ❖ Donc vrai aussi avec du logiciel libre
- ◆ Conseil, formation, installation, support, infogérance, intégration (choix), adaptation, développement, certification, migration...
- ◆ Tendances : petites sociétés --> concentration
 - ❖ Grosses sociétés ou sociétés spécialisées

Autres métiers autour du LL

◆ Hébergeurs

- ❖ Matériels, applications, informations
- ❖ Utilisation LL pour infrastructures, admin, surveillance

◆ Vendeurs de matériel

- ❖ Par exemple : Linux embarqué avec piles logicielles standard (IP, HTTP, audio-video...)

◆ Editeurs de logiciels propriétaires

- ❖ Double licence version libre – version propriétaire
 - ✓ MySQL, Zimbra...
- ❖ Utilisation de briques libres (Java, Apache...) dans des produits propriétaires

Autres métiers autour du LL

- ◆ Editeur de logiciels libres
 - ❖ Développe des logiciels avec licence libre
 - ❖ Vit de service : maintenance (MAJ), support, formation
- ◆ Editeur de distributions linux
 - ❖ RedHat, Suse, Novell, Ubuntu, Mandriva, Centos, ...
 - ❖ Intégration de très nombreux logiciels --> paquet
- ◆ Société de formation
- ◆ Métier juridique
 - ❖ Formation, conseil sur les licences, contrats...
- ◆ Agence de communication
 - ❖ Promouvoir les projets LL, organisations (fondations...)

De quoi vivent les sociétés ?

- ◆ **Ressources habituelles**
 - ❖ Service (vente de temps et d'expertise), distribution payante, hébergement...
- ◆ **Ressources liées à la gratuité et l'image**
 - ❖ Branding (teeshirt..., licence d'utilisation de la marque, franchise...)
 - ❖ Publicité, Sponsor (Google => Mozilla)
- ◆ **Subventions**
 - ❖ Bien pour tous --> subvention publique (projets de recherche européens..)
- ◆ **Double licence**
- ◆ **Extensions de LL ----> logiciels propriétaires**
- ◆ **Assurance / garantie : guichet unique pour client**

LL / économique du logiciel

- ◆ Secteur économique de plus en plus important
 - ❖ EU (2008) : LL représente 2,5 % en CA de l'info
- ◆ Evolution
 - ❖ Prévisions Gartner (mi 2008) : en 2010 Open Source 22 % du marché, 75 % des nouveaux projets
 - ❖ Assurance/garantie : principale source de revenus ?
 - ❖ Disparition des petites sociétés
- ◆ France bien positionnée (économie) mais
 - ❖ Pas une priorité pour l'Etat
 - ❖ Lobbying important de Microsoft, ORACLE, SAP...

Les LL dans les administrations

Etude DSI de l'administration été 2008

- ◆ Interviews : 20 admin centrale, 60 locale
- ◆ Utilisation LL ? : 100 % en central, 85 % en local (70 % en 2007)
- ◆ Budget LL : 10 % 2007, 13 % 2008
- ◆ Augmentation budget : prestations externes (maintenance 30 %, formation 50 %)
- ◆ En France l'administration est motrice :
 - ❖ Gendarmerie, finances, défense nationale, collectivités territoriales, certaines universités

Entreprises et administrations

Enquête LeMagIT déc 2008 – FR - 321 réponses

- ◆ Une solution LL déployée : 93 %
- ◆ Les + LL : sociétés informatique et télécoms, administration, éducation
- ◆ Les - : agro-alimentaire, BTP
- ◆ Les freins : immaturité (33%), support (25%), sécurité (17%), compétences internes (17%), dév non maîtrisé (8%)
- ◆ Support choisi : Société de service (41%), interne (36%), éditeur (23%)

Certaines entreprises ont un service logiciel libre

Quelques prestations LL

Une entreprise/administration peut

- ◆ Appels d'offre - contrats de développement spécifiant d'utiliser des logiciels libres
 - ❖ Respecter des standards
 - ❖ Demander le code source
 - ❖ Indiquer l'ayant droit pour utiliser, distribuer, revendre...
- ◆ Confier à une société
 - ❖ La maintenance et la mise à jour de son parc de logiciels libres avec une hot line
 - ❖ La maintenance d'un service (basé sur du libre) avec un délai max de rétablissement du service

Les LL utilisés en entreprise et aussi dans le secteur public

Enquête LeMagIT - déc 2008 FR - 321 réponses

- ◆ Infrastructure : OS, messagerie, gestion de données, sécurité
- ◆ Applications : bureautique (navigateurs ?), gestion de contenu (CMS)

Sources Qualipso (www.qualipso.org) avril 2008 EU

- ◆ Sécurité, internet, infrastructure (réseau, serveur de fichiers, mails, de communication, ...), bureautique, multimédia, gestion, gestion des connaissances

Types de logiciels sur PLUME (O1/2010)

- ◆ Métiers Ens et Rec (81), développement (88), internet (65), admin système (53), bureautique (42), BdD (32), multimédia (25), admin réseau (24), sécurité (21)

Les LL utilisés en entreprise

et aussi dans le secteur public

- ◆ Les LL sont encore utilisés sous cape
 - ❖ sans que la DSI le sache (et les stats)
 - ❖ par les développeurs, utilisateurs éclairés...
- ◆ Domaines que nous allons voir
 - ❖ Infrastructure : OS, admin système, admin réseau
 - ❖ Web - outils communication - travail collaboratif
 - ❖ Gestion de contenu
 - ❖ Annuaire – Base de données
 - ❖ Sécurité
 - ❖ Bureautique
 - ❖ Développement (bon exemple métier pour vous)
 - ❖ Autres

Les exemples de logiciels suivants sont tous issus de PLUME sauf
ceux en italiques

Infrastructure

◆ OS serveurs : distributions Linux

- ❖ Linux majoritaire comme serveur Web
- ❖ Et présent sur beaucoup de serveurs réseau, BdD...
- ❖ La distribution choisie dépend :
 - Du coût, du parc existant, des habitudes des administrateurs --> sites homogènes, des applis : ex : sécurité --> OpenBSD
 - *Debian, Redhat, CentOs, Fedora, Scientific linux*

◆ OS postes utilisateurs

- ❖ Fin 08 – FR : 20-30 M de PC, 1,2% Linux : 240-360 K
- ❖ Ubuntu à la mode mais aussi *fedora, mandriva, gentoo*
 - Avantage / Windows : tout sur un CD
 - Permet de réparer un disque Windows

◆ A noter Wine : exécuter prog Windows sous Unix

Infrastructure

◆ Administration système

❖ Gestion de parc

- OCS Inventory NG : inventaire dynamique (serveur+agents)
- GLPI : gestion de parc informatique avec helpdesk

❖ Déploiement, administration de n machines

- Quattor : déploiement et gestion de config de machines Linux
- cfengine : administration automatisée de systèmes hétérogènes

❖ Journalisation

- syslog-ng : gestion, centralisation de journaux informatiques

❖ Sauvegarde

- BackupPC : sauvegarde de machines Unix ou Windows

❖ ...

Infrastructure

◆ Administration système

❖ Surveillance

- Nagios : surveillance de services réseau et système
- RRDTool : stockage et visualisation de données de monitoring

❖ Partage fichiers, serveur d'impression

- Samba : partage de fichiers et d'imprimantes
- CUPS : système d'impression

❖ Virtualisation

- Xen : moniteur-hyperviseur de machines virtuelles

❖ Gestion de tickets-demands

- MANTIS : gestion des incidents (bugtracking)

❖ ...

Infrastructure

- ◆ Administration réseau
 - Idem admin système +
 - ❖ Commutateurs, routeurs --> peu utilisation logiciel libre sur PC
 - Mais Linux souvent comme OS interne ?
 - ❖ Services
 - *Bind* (DNS)
 - Asterisk : Commutateur téléphonique logiciel (ToIP)
 - ❖ Tests-debug
 - Iperf : mesure de bande passante d'un réseau IP
 - Wireshark : capture et analyse de paquets réseau
 - ❖ ...

Web

◆ Serveur Web

- ❖ Apache
- ❖ Statistiques accès
 - Awstats : serveurs web, mail...
 - phpMyVisites : stats détaillées de serveurs web

◆ Navigateur

- ❖ Firefox : 61 % accès PLUME (IE 27 %)
- ❖ Et ses 500 et plus extensions
 - Web Developer, HTML Validator, DownThemAll (gestionnaire de téléchargements), Firebug (débug JavaScript, XHTML, CSS), Lightning (agendas personnels et partagés)
- ❖ SeaMonkey : de moins en moins utilisé

Outils de communication

- ◆ **Messagerie (SMTP, POP, IMAP)**
 - ❖ Relai SMTP : *Sendmail, Postfix*
 - ❖ Serveur IMAP : Cyrus IMAP, Dovecot
 - ❖ Serveur de listes de diffusion : sympa
 - ❖ Client : Thunderbird (+ RSS), evolution (+ agenda, gestion des tâches)
- ◆ **Télé-visiophonie (sur IP):**
 - ❖ Ekiga : visioconférence, téléphonie, messagerie instantanée
 - ❖ skype non libre !

Travail collaboratif

◆ Visioconf

- ❖ VIC (video), RAT (audio) : ancêtres
- ❖ XMeeting : visioconférence et téléphone (Mac)
- ❖ VideoLAN VLC : diffusion et réception de flux multimédias : origine Ecole Centrale Paris

◆ Plateforme

- ❖ Zimbra : annuaire, messagerie, agenda partagé...
- ❖ LibreSource : Java/J2EE : wiki, zones de téléchargement, forums, formulaires, messagerie instantanée, dépôt SVN...

◆ Socle applications (développement)

- ❖ Horde : applis internet, travail coopératif...

Gestion de contenu

◆ CMS

- ❖ SPIP : articles (initialement pour journaux)
- ❖ Drupal : le CMS qui monte
- ❖ Joomla!
- ❖ Lodel : revues électroniques

◆ WIKI

- ❖ Dokuwiki (simple)
- ❖ Mediawiki (plus complexe)
- ❖ TiddlyWiki : wiki personnel sous la forme d'un seul fichier HTML
- ❖ PmWiki : wiki simple

◆ Divers

- ❖ DSpace : accès, gestion et conservation des travaux universitaires

Annuaire – Base de données

- ◆ Annuaire (LDAP)

- ❖ OpenLDAP : serveur annuaire LDAP

- ◆ Base de données

- ❖ MySQL : serveur de bases de données relationnelles
 - MySQL : société commerciale
- ❖ PhpMyAdmin
 - Gestion base MySQL à travers une interface Web
- ❖ *PostgreSQL*

Sécurité

◆ Poste travail

- ❖ Clamwin : antivirus et antispyware
- ❖ TrueCrypt : chiffrement de données

◆ Système - réseau

- ❖ MD5summer : empreintes MD5
- ❖ OpenSSH : serveur et clients SSH
- ❖ OpenVPN : connexion à distance sécurisée
- ❖ FreeRADIUS : authentification RADIUS
- ❖ J-chkmail : filtrage de courrier électronique
- ❖ mod_proxy : relais Apache
- ❖ ALCASAR : portail captif authentifiant, traces...

Bureautique

- ◆ Suite bureautique : OpenOffice.org
 - ❖ Traitement de texte : Writer
 - ❖ Tableur : Calc
 - ❖ Création de présentations : Impress
 - ❖ Dessin : Draw
 - ❖ Base de données intégrée : Base
 - ❖ Editeur d'équations : Math
 - ❖ Des dictionnaires
- ◆ Concurrent de la suite Office
 - ❖ Mêmes interfaces
 - ❖ 2 formats standards OSI : ODF et Open XML (OOXML)
 - OSI : choix technique --> lobby politico-commercial

Bureautique

◆ Fichiers

- 7-zip : compression/décompression
- Comparaison : WinMerge (fichiers et répertoires)

◆ Fichiers PDF

- PDFcreator : création
- Foxit Reader, Sumatra PDF, Xpdf : visualisation (lecteur, afficheur)
- PDF Split and Merge : découpage ou assemblage

◆ Fichiers TeX

- LaTeX : système de composition de documents
- LaTeX Beamer : classe pour créer des présentations
- Lyx : éditeur LaTeX
- TeXnicCenter : interface graphique pour TeX et LaTeX

Développement

- ◆ 60 logiciels taggués 'développement'
 - Des exemples ci-après
- ◆ Editeurs de texte (écrire du code)
 - Vim
 - JEdit
- ◆ Gestion projets (développement)
 - ❖ Ciblés logiciels
 - Subversion (SVN) : gestion de versions
 - Git : gestion de versions décentralisé
 - ❖ Généraux
 - Trac : gestionnaire de projets
 - Redmine : successeur de Trac ?

Développement

◆ EDI

- Environnement de Développement Intégré – IDE
- *Eclipse* : polyvalent, n'importe quel langage de programmation, origine IBM
- NetBeans : EDI supporté par SUN
- KDevelop : EDI pour KDE (environnement de bureau)

◆ Développement Web

- LAMP : ensemble Linux – Apache – MySQL – PHP : briques de base pour construire des sites Web
- HTML Validator : extension Firefox
- ZK : environnement de développement AJAX
- Firebug : extension Firefox : débogueur JavaScript, XHTML, CSS
- Prototype : bibliothèque Javascript de bas niveau

Développement

◆ Divers

- ❖ OpenSSL : bibliothèque chiffrement...
- ❖ Log4J : journalisation (log) des applications Java
- ❖ JUnit : développement de tests unitaires Java
- ❖ Poedit : éditeur pour traduire des programmes et des interfaces
- ❖ SCons : construction de logiciels, automatisation de la compilation (Python)
- ❖ Qt : bibliothèque création d'interfaces graphiques (GUI)

◆ ...

- ❖ <http://www.developpez.com/>

LL : Autres

- ◆ **Extrait 'top10' PLUME (non cités avant)**
 - GvSIG : Système d'Information Géographique (SIG)
 - Moodle : plateforme de formation à distance
 - Zotero : gestion de base de données bibliographique personnelle
 - JabRef : gestion de références bibliographiques et intégration des citations dans un traitement de texte
 - Xfig - WinFIG : dessin de diagrammes sous X11
 - Unitex : traitement de corpus utilisant des technologies à états finis
 - Mencoder : encodage audiovisuel multi-plateformes
 - Topcased : atelier de développement d'applications et de systèmes critiques
 - ExeLearning : création de ressources pédagogiques intégrables dans une plateforme LMS et dans un site web

LL : Autres

❖ Multimedia

- Photos, Images, Son, Vidéo

✓ www.framasoft.net

❖ Education (primaire-secondaire)

✓ www.framasoft.net

❖ PGI - ERP

- Cela semble décoller : articles, conférences...

- Cause : SAP (et les autres) : lourd et très, très cher

- Livre blanc PGI Open Source de la société SMILE :

✓ http://www.smile.fr/publications/livres-blancs/erp-open-source__1

❖ Matériels embarqués

- Pas d'existant : terrain vierge --> tout est possible

✓ Android plateforme mobile google basée sur Linux

En prenant un peu de recul...

- ◆ Un logiciel libre c'est à la fois
 - Du juridique (c'est l'origine de la définition)
 - ✓ Licences --> droit d'utilisation, de faire du commerce
 - Un objet : programme
 - ✓ Avec le code source lisible
 - Un produit fini qu'on utilise
 - ✓ Que tout le monde peut utiliser
 - Un produit pour construire d'autres produits
 - ✓ Disponible pour tous
 - Une méthode de production
 - ✓ Ré-utiliser ce qui existe, gérer des contributeurs...
 - Une économie : des sociétés qui en vivent
 - Une éthique de vie :
 - ✓ Contributeurs et communauté : vivre ensemble
 - ✓ Contributions gratuites : donner

En prenant un peu de recul...

- ❖ Bcp d'objectifs orthogonaux --> équilibre
 - Les sociétés de service reversent leurs développements à la communauté
 - Les fondateurs veillent aux dérives : dénonciations, alertes médiatiques, documents de re-centrage
- ❖ Aide pour les pays pauvres ?
- ❖ Des faits
 - Croissance de leur utilisation
 - ✓ Sont maintenant reconnus comme des logiciels professionnels
 - ✓ Sont présents dans les logiciels généraux et dans tout ce qui est innovant
 - (R)évolution pour les professionnels de l'informatique
 - ✓ Reprise de pouvoir de la technique ?
 - ✓ Il y a des pour et des contres.
 - ✓ ATTENTION : certaines DSI complètement opposées : perte de pouvoir ?

Remerciements

A tous les auteurs de présentations de
l'école ENVOL en particulier

Thierry Aimé, Matthieu Herrb, Arnaud
Laprevote, François Miller, Cédric Thomas

<http://www.projet-plume.org/fr/ressource/presentations-envol-2008>

A tous les auteurs de fiches PLUME en
particulier Geneviève Romier, Teresa
Gomez-Diaz, Violaine Louvet

<http://www.projet-plume.org/fr/contributeurs>

Références

Ce sont plutôt des points d'entrée pour avoir accès à d'autres documents

- ❖ Pour diffuser en libre : licences, copyright, forge...
 - ✓ <http://www.projet-plume.org/fr/ressource/referentiel-doc-dev>
- ❖ Communautés, XP, forge, gestion de version, écosystème
 - ✓ <http://www.projet-plume.org/fr/ressource/presentations-envol-2008>
- ❖ Le LL dans l'Ens Sup, en France, dans le monde
 - ✓ http://www.projet-plume.org/ressource/Logiciel_Libre_dans_ESR_et_ailleurs
- ❖ Description de logiciels
 - ✓ Les fiches descriptives PLUME
<http://www.projet-plume.org/fr/fiches-documents>
- ❖ Informations sur les logiciels libres
 - ✓ Ressources PLUME
<http://www.projet-plume.org/fr/ressources>