



Transfert et logiciels libres à l'INRIA

Patrick MOREAU
Responsable du Patrimoine Logiciel
Direction du Transfert et de l'Innovation

patrick.moreau@inria.fr

Paris , le 22 mai 2012

Introduction

Stratégie Inria sur l'open source

Une analyse du processus de passage du Floss de recherche à du Floss commercial

Quelques exemples concrets

Pourquoi un responsable patrimoine logiciel?

En terme de dynamique de maturation, le logiciel est radicalement différent du brevet.

Le logiciel est un objet beaucoup plus plastique et dynamique

- avec une imbrication forte entre la production et le transfert (rôle de la demande) :

Les éléments qui caractérisent la maturation du logiciel sont ainsi :

- **la non autonomie de l'actif technologique**, qui est directement lié à la production scientifique de R&D (alors que le brevet est identifiable de manière externe et autonome) ;
- **la durée de la maturation**, les logiciels pouvant résulter d'un long cycle de développement/accumulation (ils peuvent constituer de fait une « grappe ») ;
- **l'éclatement potentiel des contributions** (droit d'auteur, nombreux contributeurs, cas spécifique de l'Open Source, rôle croissant du « web » avec la réutilisation de codes préexistants).
- **parfois couple compétence/actif techno** (= création d'entreprise souvent un bon vecteur).

Nécessité d'avoir une approche spécifique du transfert par le logiciel

Les dimensions du Logiciel Libre

Le Logiciel Libre recouvre plusieurs dimensions :

- objet de recherche
- objet de diffusion scientifique
 - Open source de recherche
- objet pour le monde industriel, les éditeurs ou le service
 - Open source commercial
- outil de transfert
 - Open source de transfert
 - « Valorisation » car création de valeur

Objet de Transfert



Le Logiciel Libre est un objet complexe et le choix de ce mode de diffusion / transfert doit être mûrement réfléchi afin de rechercher la stratégie qui maximisera l'impact sur la société (scientifique, sociétal et économique)

Il se peut qu'il y ait parfois une contradiction entre les logiques de diffusion scientifique et de transfert

- utiliser systématiquement la diffusion des résultats sous forme de Logiciel Libre peut obérer le potentiel de transfert de la recherche publique
- le modèle du Logiciel Libre n'a pas toujours l'impact escompté y compris en terme académique

Le Logiciel Libre ne doit pas être motivé par: « I code, I publish, I put the software on the web, I exist »

Choix de la licence: un choix optimal

C'est un élément primordial qui nécessite une vraie réflexion.

La licence envisagée dépend (entre autres):

- de la nature (bibliothèque, framework)
- des intentions d'exploitation
 - qu'est ce que je cherche à faire?
- de la cible, de l'impact
- de la modularité du logiciel
- de la facilité pour un contributeur externe de comprendre comment contribuer
 - Barrière techno, existence de doc, facilité de tests, ...
- de l'étendue des droits que l'on souhaite accorder aux tiers en termes de distribution
- Du temps à passer pour développer la communauté

Nécessite des discussions avec son « chargé de valorisation »

Tableau de synthèse

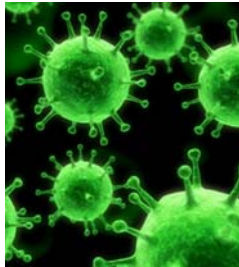
Nature	Catégorie	Exemple de licence	DERIVATION Modification et redistribution ?	COMPOSITION Agrégation et redistribution ?
NON - LIBRE	"Propriétaire"		N'est pas automatique	N'est pas automatique
LIBRE	non permissive en dérivation et composition	CeCILL, GNU GPL	sous la même licence	sous la même licence
NON PERMISSIVE	non permissive en dérivation, permissive en composition	CeCILL C, GNU LGPL	sous la même licence	OUI, y compris sous une licence différente
LIBRE PERMISSIVE	en dérivation et composition	CeCILL B, BSD	OUI, y compris sous une licence différente	OUI, y compris sous une licence différente

Des termes entendus ... et à ne plus entendre

Copyleft

Permissif / non-permissif

~~Viral~~



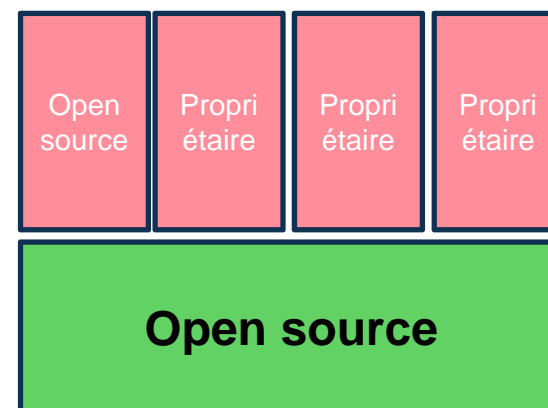
~~Contaminant~~



Alternative au Logiciel Libre

Tout ce qui n'est pas Libre est non-Libre mais il existe une large variété de mode d'exploitation

- Licence ad-hoc pour le partage scientifique
 - exécuter le programme
 - l'étudier
 - l'adapter
 - Pas le droit de le redistribuer.
- Les modèles mixtes
 - Double licence
 - License décalée (ou chronodégradable)
 - Associations de modules libres avec des modules propriétaires



Stratégie d'utilisation du Logiciel Libre

La définition de la stratégie d'utilisation du Logiciel Libre comme un outil de transfert conduit à deux axes de réflexions et d'actions:

- internes:
 - Professionnalisation des pratiques d'édition
 - Accroissement de l'impact des logiciels développés et stratégie de sortie
- externes:
 - Contribution à la création de la confiance autour du Logiciel Libre
 - Mise en place d'écosystèmes facilitant le transfert et l'innovation

Actions internes Inria

Rédaction d'un document de préconisation des choix (qualification) illustrés de retour d'expériences

Etablissement de la check list décisionnelle pour le choix de diffuser le logiciel sous une licence libre

Dégager et identifier des moyens humains et financiers pour l'industrialisation du logiciel et, le cas échéant, la constitution et l'animation de communautés

Mise en œuvre d'une méthodologie outillée d'analyse IPR (Intellectual Property Rights)

- Confiance dans les logiciels

Evaluer et qualifier le patrimoine Logiciel Libre de l'INRIA

Définition d'indicateurs de mesure d'impact du Logiciel Libre

Actions externes Inria

Participer au lancement de l'IRILL (Oct 2010)

Diffuser la position de l'INRIA dans l'écosystème

Conforter le positionnement de l'INRIA dans le contexte de mutualisation du transfert en France (best practices)

Rendre lisible la production agrégée de Logiciel Libre de l'INRIA

Créer un observatoire national du Logiciel Libre afin d'élargir les mesures d'impact

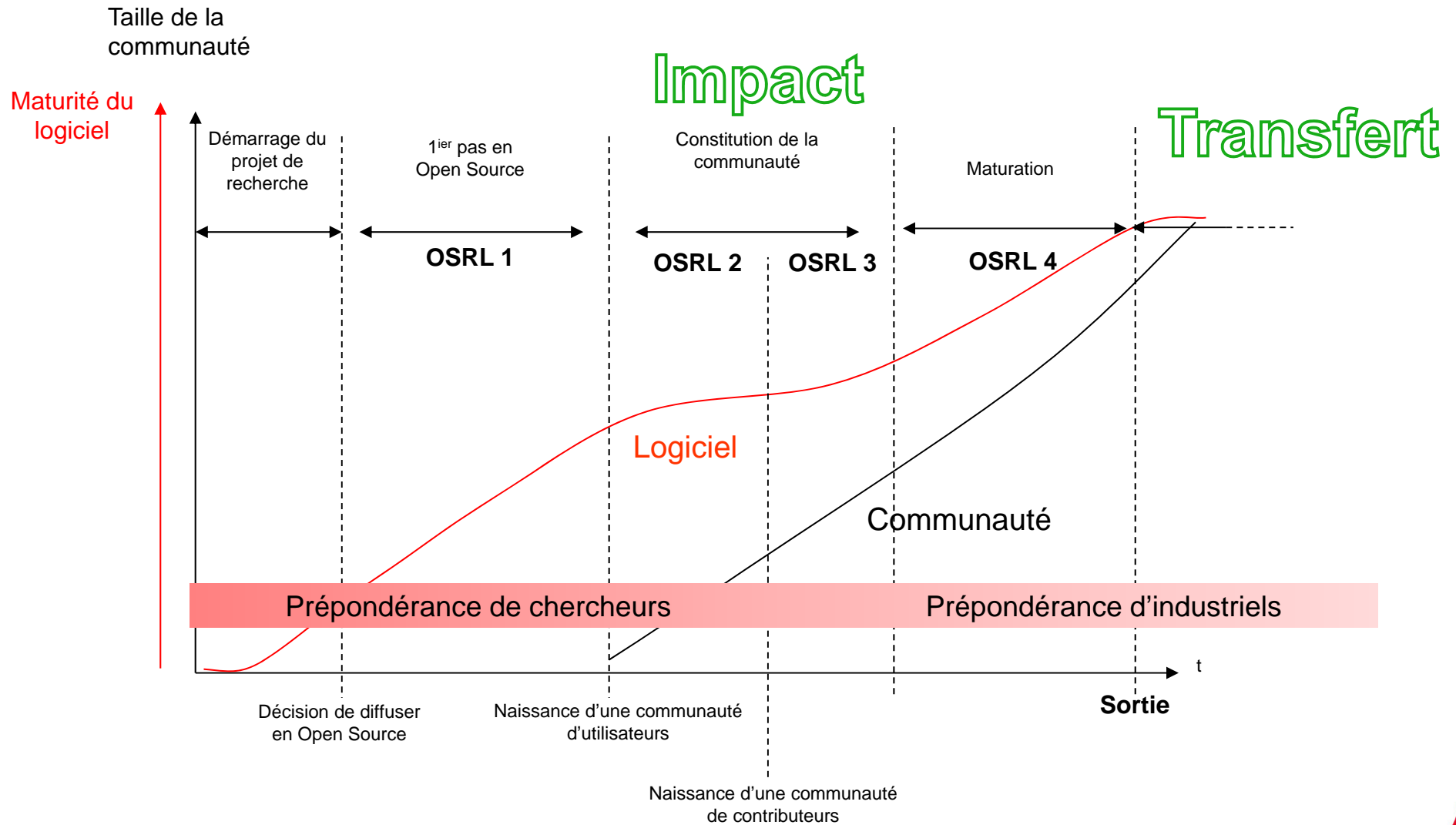
La communauté

C'est un moyen, pas une fin!

Un premier exemple:

- Un gros logiciel open source dans un centre INRIA
- Décision de création de start-up sur un schéma de double licensing
- La licence permissive du logiciel n'est alors plus adaptée
 - Décision de changement de licence
- 6 mois de travaux pour rendre compatible le logiciel avec la licence choisie

Cycle de vie idéal de l'open source de recherche



Cycle de vie idéal de l'open source de recherche

Ce cycle de vie idéal peut se découper en 4 niveaux que nous appelons OSRL (Open Source Readiness Level) :

- OSRL 1 : juste après la diffusion, pas d'utilisateurs externes,
- OSRL 2 : une communauté d'utilisateurs commence à voir le jour,
- OSRL 3 : une communauté de contributeurs se développe,
- OSRL 4 : la communauté se consolide et le processus de maturation se met en place

La durée de ce cycle est très longue

- 10 ans en moyenne
- La communauté peut évoluer au long de ce cycle

Et les efforts sont importants

Conditions générales

Il existe des bonnes pratiques à mettre en œuvre quel que soit la phase du cycle et quel que soit le projet de développement logiciel dans la recherche publique :

définition d'une stratégie de diffusion/d'exploitation du logiciel dès que possible, avec une vision long terme,

- mise à jour de la stratégie précitée tant que de besoin,
- bonnes pratiques de développement, en accord avec la stratégie autour du logiciel (notamment le respect des compatibilités de licence),
- stratégie juridique adéquate, en accord avec la stratégie autour du logiciel,
- bonne gestion des relations entre les copropriétaires, la majorité des logiciels libres de la recherche publique étant développés par des équipes relevant de plusieurs établissements différents.

OSRL 1 : les premiers pas

Le type de public visé doit être identifié.

- Public averti ou non
- Si l'on s'adresse à un public averti, la documentation peut être succincte, et ne pas détailler certains aspects.
- Au contraire, pour un logiciel où le public est novice en informatique, les activités liées au support sont primordiales : la documentation doit être très précise, des tutoriels doivent être mis en ligne, les mailings list doivent être entretenues quotidiennement.

OSRL 1 : les premiers pas

Mise en place des outils nécessaires à la constitution d'une communauté

- Des mailing list (utilisateurs et développeurs), pour pouvoir communiquer avec les membres de la communauté, annoncer des releases, signaler des événements....,
- Le bug tracker est également nécessaire, il permet de signaler des bugs de toute nature.
- L'accès du logiciel par une forge.
- Une documentation doit accompagner les différents releases et, doit être mise à jour régulièrement.
- Un site web composé de :
 - Une présentation du projet.
 - Un espace de discussion en ligne.
 - Un historique des releases.

OSRL 1 : les premiers pas

Il est conseillé qu'une personne, idéalement un chercheur permanent, suive le projet de près afin de maîtriser l'architecture du code, et pouvoir guider les futurs utilisateurs dans l'utilisation du logiciel.

La motivation du porteur du projet à créer une communauté autour de son logiciel est alors cruciale.

Pour faciliter une large diffusion, il est conseillé de porter le logiciel sous différents systèmes d'exploitation (Windows, Linux, Mac Os...).

- Si le logiciel s'adresse à un public novice, un lien d'installation doit être mis à disposition.

OSRL 1 : les premiers pas

Il est important de faciliter l'ajout de fonctionnalités en standardisant les interfaces et en adoptant une architecture modulaire.

- Aussi, le fait de proposer une version stable et une version en développement facilite l'utilisation du logiciel par des tiers.

OSRL 2 : constitution d'une communauté d'utilisateurs puis OSRL 3 : constitution d'une communauté de contributeurs

Les activités de ces deux phases sont relativement similaires, certains utilisateurs devenant au fur et à mesure des contributeurs.

- Rares sont les contributeurs développeurs autour de l'open source de recherche.

OSRL 2 : constitution d'une communauté d'utilisateurs puis OSRL 3 : constitution d'une communauté de contributeurs

Organisation des retours de la communauté

- Décision d'intégrer ou non, de tenir compte des suggestions et des retours d'expérience de la communauté, en fonction de leur intérêt et de leur pertinence.
- Décision également d'intégrer ou non les contributions directes au logiciel (spécifications détaillées, code...), s'il y en a, et effectuera ces intégrations.
- Si les contributions sont nombreuses, il est recommandé de favoriser la création d'une hiérarchie parmi les contributeurs afin de déléguer ces tâches, ceci en fonction des parties du code visées (noyau versus modules)
 - Dans la pratique, cette hiérarchie est souvent liée à la méritocratie.
- Pensez le cas échéant aux cessions de droit

OSRL 2 : constitution d'une communauté d'utilisateurs puis OSRL 3 : constitution d'une communauté de contributeurs

Animation de la communauté

- Marketing et communication
- il est conseillé de sortir régulièrement des versions et de les annoncer sur une liste de diffusion.

Assurer l'évolution de la communauté :

- Ceci peut passer par l'évolution de la gouvernance de la communauté ; qui peut parfois devenir plus formalisée.

Difficultés identifiées :

- Au cours de ces phases, le développement du logiciel est en général ralenti.
- En effet, les porteurs de projet concentrent leurs efforts sur la coordination et l'animation de la communauté, et ont moins de temps pour développer le logiciel.
- Remarque : une attention particulière doit être portée pour qu'il n'y ait pas de rupture dans la connaissance du logiciel et de son architecture, mais aussi de la connaissance de la communauté et de son fonctionnement, malgré des changements de ressources humaines, inévitables sur des projets de longue durée.

Exemple: Coccinelle

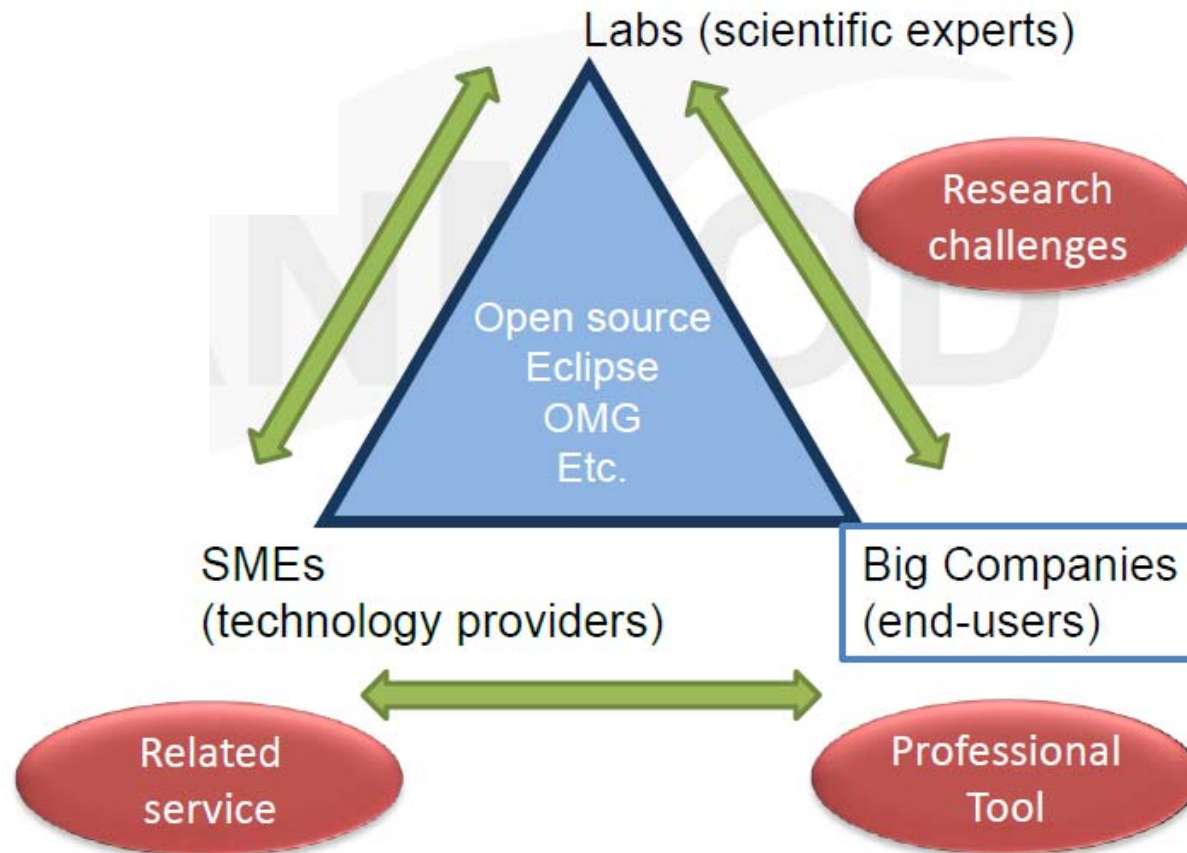
Sep 2010: Les travaux de recherche du projet Coccinelle ont été dans le noyau du système d'exploitation Linux

- Recherche des bugs au moyen de "patches sémantiques".
- Pas moins de 600 correctifs, issus de patches sémantiques, sont intégrés à la version de développement du noyau Linux.

La démarche

- acceptation par les développeurs Linux qui ne sont pas des experts du domaine de recherche
- projet de longue haleine: démarrage en 2004/2005. Le 1er prototype est sorti en 2007

Exemple: ATLANMOD



© AtlanMod - atlanmod-contact@mines-nantes.fr

Exemples: Sans nom 😊

Logiciel qui n'est plus un objet de recherche

Départ en retraite des leaders

« La mayonnaise prend » mais la gouvernance n'est pas lâchée

- Entre 2 eaux

Merci pour votre attention

Merci

Inria