

**Valorisation du Logiciel Libre
au Grand-Duché de Luxembourg :
les modèles d'exploitation
de l'informatique libre**

Alexandre Dulaunoy, Matthieu Farcot, ...

Dans le cadre du projet FEDER : AIDILL (Animation, Information et Développement de l'informatique Libre au Luxembourg) mandaté par le Ministère de l'économie du Grand-Duché de Luxembourg.

© Copyright 2003 by Alexandre Dulaunoy, Matthieu Farcot, ...

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Dédicace

A la communauté du Logiciel Libre au Grand-Duché de Luxembourg et partout ailleurs.

Remerciements

Ce document est le résultat d'un travail collaboratif réalisé par Alexandre Dulaunoy, Matthieu Farcot, Christophe Feltus, ... (ajouter les personnes qui ont participé à la rédaction de ce document)

Merci au Centre de Recherche Henri Tudor (CRP-HT) pour le cadre du projet AIDILL.

Merci au Ministère de l'économie du Grand-Duché de Luxembourg pour ce projet FEDER.

Merci à Donald Knuth for \LaTeX et tous les membres actifs de la communauté du libre...

Terminologie et conventions typographiques

Des noms éphémères sont utilisés pour décrire des entités fictives du paysage luxembourgeois :

Foobar s.à.r.l.	Une PME luxembourgeoise dans le logiciel avec une activité de support informatique de 20 personnes
Baz s.a.	Une entreprise luxembourgeoise dans les télécommunications
L-Grounds s.a	Une entreprise de service et création de logiciel propriétaire
Fred & Barney law firm	un cabinet d'avocat luxembourgeois spécialisé dans le droit d'auteur
Université QuuX	Université et centre de recherche luxembourgeois
Alice et Bob	Deux citoyens luxembourgeois
Carol et Henry	Deux citoyens non-luxembourgeois

La terminologie GNU/Linux sera utilisée pour décrire le système d'exploitation informatique complet (par exemple la distribution Debian GNU/Linux). Le terme Linux sera utilisé uniquement pour décrire le noyau et les activités relatives à ce noyau.

*L'italique dans une
boite*

signifie un exemple concret pratique de la théorie présentée.

Table des matières

Dédicace	iii
Remerciements	iv
Terminologie et conventions typographiques	v
Abstract	x
1 Introduction	1
1.1 Logiciel Libre	1
1.1.1 Historique du logiciel libre	4
1.1.2 Le libre au-delà du logiciel	6
1.2 Le Grand-Duché de Luxembourg	7
1.2.1 Acteurs du Logiciel Libre	7
1.2.2 LiLux	7
1.2.3 Linuxdays (CRPHT)	7
1.2.4 Projets libres initiés au Luxembourg	7
1.3 Ligne directrice du rapport	7
2 Economie et Logiciel Libre	8
2.1 Approche de l'industrie du logiciel	8
2.1.1 Industrie du logiciel	8
2.1.2 Innovateurs et innovation, un intérêt convergeant ?	11
2.1.3 Nature des licences et logiques de diffusion	13
2.1.4 Tendances	16
2.1.5 Logiciel, produit et cycle de vie	17
2.1.6 Logiciel fermé, logiciel libre	19
2.2 Caractérisation des utilisateurs	21
2.2.1 Pragmatiques et idéalistes	21
2.3 Axe Demande	22
2.3.1 Demande en logiciel propriétaire	22

2.3.2	Demande en logiciel libre	22
2.4	motivations	22
2.4.1	Administrations	22
2.4.2	Firmes dont le développement est une activité primaire	24
2.4.3	Firmes dont le développement est une activité secondaire	24
2.4.4	Individus	25
2.5	Innovation et succès	25
3	Modèles d'exploitation du logiciel libre	27
3.1	Entreprises	27
3.1.1	Utilisation	27
3.1.2	Développement	27
3.2	Citoyens	27
3.2.1	Utilisation	27
3.2.2	Développement	27
3.3	Exemples existants de modèles d'exploitation	27
3.3.1	Externaliser un logiciel	27
3.3.2	Partager le développement	27
3.3.3	Produire des produits connexes	27
3.3.4	Mode dualiste : libre et propriétaire	28
3.3.5	Mode dualiste séparé : libre et propriétaire	28
3.3.6	Mode services libres + logiciels propriétaires	28
3.3.7	Mode libre - fondation	28
4	Application des théories économique	29
5	Le cadre légale	30
5.1	Licences logicielles Libres	30
5.1.1	Les licences libres "copyleft"	31
5.1.2	Les licences libres "non-copyleft"	32
5.1.3	Les licences non-libres	32
5.1.4	Comment appliquer une licence libre ?	32
5.2	Licences libres et les obligations contractuelles	33
5.3	Licences libres non-logiciels	33
5.4	Logiciel libre et brevet	33
6	Conclusion	34
A	GNU Free Documentation License	35

Biographical Note 47

Table des figures

2.1	Phases du développement logiciel	9
2.2	Développement non linéaire et Propriété Intellectuelle	11
2.3	Licences et droits d'usages	13
2.4	Licences et droits d'usages	14
2.5	Cycle de vie d'un produit	17
2.6	Cercle de développement, logiciel propriétaire	19
2.7	Cercle de développement, logiciel libre	20
2.8	Cycle de vie du logiciel libre	20
2.9	Populations et motivation	21
2.10	Communication mono-directionnelle	22
2.11	Communication bi-directionnelle	22

Résumé

Valorisation du Logiciel Libre au Grand-Duché de Luxembourg : les modèles d'exploitation de l'informatique libre

Alexandre Dulaunoy, Matthieu Farcot, ...

URL : <http://www.aidill.org.lu/>

Le logiciel libre est devenu incontournable dans l'industrie informatique et représente une alternative au logiciel propriétaire. Le Grand-Duché de Luxembourg n'y échappe pas et le logiciel libre peut être un vecteur de dynamisme local ainsi bien pour les entreprises que pour le citoyen. Ce rapport couvre les différents aspects (économique, légale et sociétale) du logiciel libre et répond aux questions les plus courantes autour du libre. Le but est sa mise en valeur pour la création d'une locale de l'innovation et de la participation collective entreprises à ce mouvement.

Chapitre 1

Introduction

1.1 Logiciel Libre

La définition du logiciel libre est énoncée[?] par la Free Software Foundation et permet de qualifier exactement ce qu'est un logiciel libre. Cette définition est composée de 4 libertés :

- La liberté d'exécuter le programme, pour tous les usages (liberté zéro).
- La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté une). Pour ceci l'accès au code source est une condition requise.
- La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté deux).
- La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté trois). Pour ceci l'accès au code source est une condition requise.

Un programme est un logiciel libre si les utilisateurs¹ possèdent **toutes ces libertés**.

La définition d'un logiciel propriétaire peut être qualifiée par rapport au nom respect des 4 libertés du logiciel libre. C'est à dire qu'un logiciel propriétaire est un logiciel qui ne respectent pas une ou plusieurs de ces 4 libertés.

Toutes ces libertés accordées au logiciel libre peuvent sembler étranges dans un premier temps mais ces différentes libertés apportent des plus indéniables par rapport au logiciel propriétaire.

¹La terminologie utilisateur englobe les personnes physiques mais aussi les personnes morales comme les entreprises ou les administrations publiques. Le terme utilisateur sera utilisé dans ce sens large.

La liberté zéro qui concède à l'utilisateur la liberté d'exécuter le programme pour n'importe quelle usage sans aucune restriction. C'est à dire que l'exécution n'est pas limitée à un nombre de machine ou un système ou une version donnée ce qui est souvent le cas du logiciel propriétaire.

Un exemple pratique de la liberté "zéro" : *Vous avez un logiciel propriétaire qui ne permet que l'exécution sur un 1 CPU mais vous disposez d'un serveur disposant de 4 CPU, vous devez acquérir (et donc acheter) une licence complémentaire pour ce logiciel propriétaire. Par contre avec un logiciel libre, ce type de restriction n'existe pas puisque vous pouvez exécuter le logiciel pour tous les usages.*

La liberté une permet d'étudier le fonctionnement d'un logiciel libre et d'apporter des modifications suivant vos besoins. C'est à dire que vous pouvez lire la recette qui compose un logiciel et comprendre toutes les interactions du logiciel. Bien entendu, l'utilisateur ne dispose toujours des connaissances pour analyser le code source du logiciel mais vous pouvez toujours demander une aide extérieur pour comprendre le fonctionnement exact d'un logiciel libre. Mais aussi vous pouvez modifier librement et de façon privée le logiciel suivant vos besoins spécifiques.

Un exemple pratique de la liberté "une" : *Vous disposez d'un logiciel propriétaire qui possède une erreur très spécifique et le rend instable lors d'une utilisation prolongée. Vous demandez à l'éditeur la correction de cette erreur mais vous devez attendre la nouvelle version disponible dans 6 mois. Par contre avec un logiciel libre, vous avez la possibilité de faire la modification vous-même et sans aucun délais.*

La liberté deux vous permet de distribuer librement des copies d'un logiciel libre. C'est à dire que vous pouvez copier et distribuer sans limite un logiciel de ce type contrairement au logiciel propriétaire où la copie et la distribution sont très souvent limités.

Un exemple pratique de la liberté “deux” *Vous travaillez chez Baz s.a. et un de vos collègues vous demande une copie du logiciel de base de données MySQL. Comme ce logiciel est libre, vous pouvez le copier librement sans aucune limite et ainsi rendre service à votre collègue.*

Un autre exemple, la société Baz s.a. développe un produit pour ses différents clients mais utilise le serveur web Apache comme base. Apache étant libre, Baz s.a. peut inclure librement le logiciel dans son package de distribution. Si elle avait utilisé un logiciel propriétaire, elle aurait dû obtenir une licence de distribution spécifique et donc, ajouter des coûts complémentaires à la distribution.

La liberté trois vous permet de distribuer les modifications réalisées à un logiciel libre et ainsi de partager les modifications avec plusieurs personnes (entreprises, citoyens, associations, organismes publics). C’est souvent la liberté la plus visible car elle concerne l’activité de travailler collaborativement à la création de logiciels.

Un exemple pratique de la liberté “trois” *L’entreprise Baz s.a. désire réaliser un logiciel en partenariat avec une autre société car elle ne dispose pas de toutes les compétences et ressources nécessaires. La Liberté trois permet de créer un logiciel libre entre ces deux sociétés de façon précise (en utilisant les techniques légales et pratique du logiciel libre) et équitable (les deux entreprises sont sur le même pied).*

Le succès actuel du logiciel libre fonctionne sur la combinaison de ces 4 libertés. Cette cohésion est importante car elle permet de garantir l’équité entre les utilisateurs (dans le sens général) en partageant les mêmes règles et principes sur le travail communautaire ² de la création de logiciels libres.

Le droit d’auteur est utilisé pour l’application de ces 4 libertés sous la forme de licences libres. Elle décrivent ces libertés sous la forme d’un “contrat”³. Ces licences libres sont les bases légales de l’existence du logiciel libre mais aussi de sa pérennité. Il existe plusieurs licences libres, les plus connues sont la GNU General Public License (ref), la licence de

²dans le sens d’une ou plusieurs interactions pour la création d’un logiciel libre.

³La terminologie exacte est une licence logicielle. Il existe des différences subtiles entre les deux.

type “Berkeley” (ref) ou la Mozilla Public License (ref).

Si vous désirez approfondir le sujet des licences du logiciel libre, le chapitre s’y rapportant 5 détails les différentes licences.

Les différentes licences libres

Les différentes licences libres sont abordées en détail dans le cadre du chapitre 5.

1.1.1 Historique du logiciel libre

Le logiciel libre existe depuis les débuts de l’industrie informatique mais il n’a commencé à exister dans sa forme connue à partir des années 80.

Le projet GNU

Au début des années 80, une majorité des logiciels étaient des logiciels propriétaires et donc les propriétaires de logiciels limitaient et interdisaient la coopération entre les développeurs mais surtout les utilisateurs.

C’est pour cela que Richard Stallman a conçu le projet GNU en 1983. Ce projet a développé un système complet de logiciels GNU (GNU’s not unix) avec une compatibilité avec les systèmes Unix. En fait, chaque ordinateur a besoin d’un système d’exploitation pour fonctionner et le but majeur du projet GNU était d’écrire un système d’exploitation libre. Après 20 ans d’existence, le projet a aussi des logiciels libres qui vont au-delà du système d’exploitation et sont utilisés par des utilisateurs. Par exemple, le projet Gnome⁴ qui fournit une interface graphique cohérente pour les utilisateurs fait partie du projet GNU. La partie manquante au projet GNU était le coeur du système d’exploitation le noyau (kernel), le projet GNU Hurd avait pour but de créer un noyau pour utiliser les composantes du projets GNU. Mais Linus Torvalds combla ce manque en réalisant un noyau libre dénommé : Linux. C’est ainsi que le projet GNU peut se pourvoir d’un système d’exploitation dénommé GNU/Linux.

⁴<http://www.gnome.org/>

Le noyau Linux

En 1991, Linus B. Torvalds décida d'écrire le noyau d'un système d'exploitation libre sous la licence GNU General Public License. Au début, la tâche pouvait sembler difficile et hasardeuse suite aux échecs successifs du noyau Hurd. L'avantage majeur du noyau Linux fut de rester simple mais d'utiliser une technologie assez bien connue (les noyaux monolithiques). Ce qui permit aussi d'élargir le cercle de développeurs par la simplicité d'accès au code source. En quelques années, le noyau a commencé par une phase très instable pour devenir un noyau de système d'exploitation les plus stables.

Les distributions GNU/Linux

Une distribution libre est un ensemble de logiciels libres comprenant le système d'exploitation et souvent d'autres logiciels nécessaire à une utilisation quotidienne de l'informatique. Une distribution GNU/Linux se base sur le noyau Linux pour le système d'exploitation et une panoplie d'outils GNU (le compilateur gcc, les bibliothèques systèmes, ...) mais aussi d'autres logiciels libres (Perl, OpenOffice, Evolution, ...).

On peut distinguer deux grandes catégories de distributions GNU/Linux. Les distributions maintenues par des entreprises commerciales comme par exemple RedHat Linux, SuSE, Mandrake,... et les distributions maintenues par une communauté de développeurs comme par exemple Debian GNU/Linux ou Gentoo. Certaines distributions sont hybrides : Il existe des distributions maintenues par des sociétés commerciales mais utilisant des distributions communautaires comme base de travail. L'exemple Progeny Linux⁵ qui utilise la Debian GNU/Linux comme base permettant de garder une certaine base de compatibilité. On peut aussi citer l'exemple de Fedora⁶ lancé par la société RedHat dans le but de construire une distribution communautaire qui servirait de base à ses propres systèmes.

Il existe aussi une multitude de distributions pour des utilisations précises comme pour l'éducation, les machines pour le calcul par grappe, pour les sauvegardes systèmes ou les applications serveurs. Pour une liste exhaustive, le site de LWN dispose d'une liste de 300

⁵<http://www.progeny.com/>

⁶<http://fedora.redhat.com/>

distributions GNU/Linux : <http://lwn.net/Distributions/>.

Applications serveurs

Les utilisations de l'informatique libre sont assez larges mais sa première application est de donner des services aux utilisateurs et donc de fournir des serveurs d'application. Historiquement, un des grands succès est l'utilisation du serveur Web/HTTP Apache (<http://www.apache.org/>) sur des systèmes d'exploitation libres. Il est un système des plus répandus sur Internet avec près de 60% de part de marché. Le logiciel libre est largement utilisé sur Internet pour tous les services Internet, par exemple, l'infrastructure de Google utilise exclusivement GNU/Linux.

Applications utilisateurs

L'informatique libre ne se limite plus aux applications pour les serveurs d'entreprise mais propose une gamme d'outils informatiques libres pour les plateformes libres (comme GNU/Linux) mais aussi propriétaires (comme Windows). C'est à dire des applications pour l'utilisation journalière de l'informatique comme des navigateurs Internet, des suites bureautiques, des outils graphiques,...

OpenOffice

Une description d'OpenOffice et de son cadre et de l'évolution rapide du projet.

Mozilla InkScape Gimp

1.1.2 Le libre au-delà du logiciel

Wikipedia, Opencores et ... montrer la dynamique et peut-être des pistes pour les exploitations

1.2 Le Grand-Duché de Luxembourg

1.2.1 Acteurs du Logiciel Libre

1.2.2 LiLux

1.2.3 Linuxdays (CRPHT)

à fournir ? autre truc CRPHT ?

1.2.4 Projets libres initiés au Luxembourg

OpenSST et autre ?

1.3 Ligne directrice du rapport

Logiciel Libre et économie locale (-> conclusion)

Chapitre 2

Economie et Logiciel Libre

2.1 Approche de l'industrie du logiciel

2.1.1 Industrie du logiciel

Quelles sont les particularités de l'industrie du logiciel ? De l'approche "propriétaire" à l'approche "libre", quelles incidences ? Comment caractériser les spécificités sectorielles de l'industrie du logiciel ?

Cycles de développement des produits

Il est important d'introduire, à destination des lecteurs ayant peu de connaissances du vocabulaire informatique technique, une brève présentation des différentes spécificités du développement logiciel.

Un logiciel consiste, pour simplifier, en un programme informatique contenant une liste d'instructions destinées à la gestion et l'exécution de certaines données. Le logiciel est un produit issu d'une création de l'esprit, immatériel, qui comprend des programmes, procédures, guides et documentations nécessaires pour donner des instructions à un outil informatique et lui faire réaliser des tâches précises.

Le développement logiciel typique comporte 4 phases distinctes (approche chronologique) :

- Le design du code, reflet des algorithmes employés, des méthodes, des interactions et de l'architecture des différents composants à mettre en oeuvre dans le cadre du fonctionnement du logiciel (représentation UML, ...).
- Le code source, issu de la programmation dans un langage particulier, exprimé sous

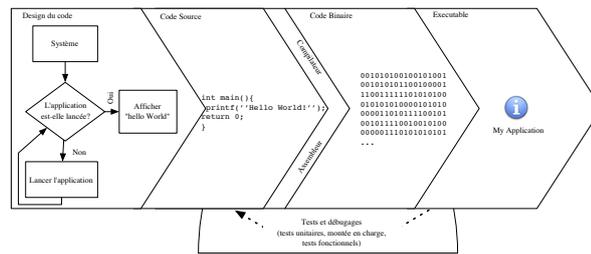


FIG. 2.1 – Phases du développement logiciel

forme lisible. Ce code source est la séquence d'instructions d'un logiciel et ce qui en détermine ses fonctionnalités.

- Le code binaire, après l'interprétation effectuée par un compilateur, destiné à rendre le code lisible par la machine. De ce code est issu l'exécutable, application sous sa forme iconographique (Graphical User Interface) ou textuelle (Command Line Interface), représentant le contact entre la majorité des utilisateurs du logiciel et le programme informatique, et intégrant les spécificités du système sur lequel il fonctionne (charte graphique, ...).
- Le phénomène de retro-action propre au débugage et tests du code, modifiant le code source de manière à élever la qualité de l'executable final.

Ce processus de développement peut trouver de nombreuses analogies. En particulier, une comparaison p

Nature physique du bien produit

Pour affiner cette lecture des cycles de développements dans l'industrie du logiciel, il est important de garder à l'esprit la distinction entre design, code source et code binaire.

Si la relation entre le code binaire, le code source et le design est très souvent vérifiée (dans cet ordre), l'ordre inverse n'est pas vrai. Pour quelle raison ? Parce qu'un design précis peut donner naissance à de nombreux codes sources différents, et que ces codes sources, en fonction des outils de compilation utilisés, vont eux aussi engendrer un code binaire particulier.

La nature physique du bien produit reste, bien évidemment, intangible (sauf dans les cas extrêmes des logiciels spécialisés dis "embarqués", dépendant d'une couche matérielle spécialisé).

Ce constat d'intangibilité de la nature intrinsèque du logiciel en tant que produit se doit d'être affiné.

Les sciences économiques ont pour usage d'appréhender la problématique de la caractérisation de la nature des biens au-travers d'une double approche :

- la propriété de rivalité et
- la propriété d'exclusivité

La propriété de *rivalité* se rapporte à tout bien qui "ne peut être extensible sans perte intrinsèque de ses qualités, de manière à ce qu'il ne puisse être utilisé par autant de gens qui le souhaiteraient" ?. Un bien rival entraîne ainsi automatiquement un effet de nuisance d'encombrement à son usage.

L'*exclusivité* pour sa part se rapporte à la notion d'appropriation. Un bien exclusif, donc démontrant des propriétés d'exclusivité, ne peut être utilisé que par un (ou plusieurs) utilisateur(s) identifié(s).

La notion de rivalité se rapporte à la notion d'action. Celle de l'exclusivité à la notion de diffusion.

Cette distinction est illustrable par le tableau suivant :

Nature du bien	Non-Exclusif	Exclusif
Non-Rival	<i>Bien Public</i>	<i>Bien de Club</i>
Rival	<i>Bien Partagé</i>	<i>Bien Privé</i>

TAB. 2.1 – Nature des biens

Une idée est non-rivale par définition. Donc une collection d'idées, regroupée sous la forme d'une oeuvre de l'esprit,

Protection et innovation

Le modèle classique d'appréhension de l'acte d'innovation, quelle que soit la nature de celle-ci, repose sur le lien généralement accepté

Inventeur → *crée Invention* → *donne Innovation* → *réussie Diffusion*

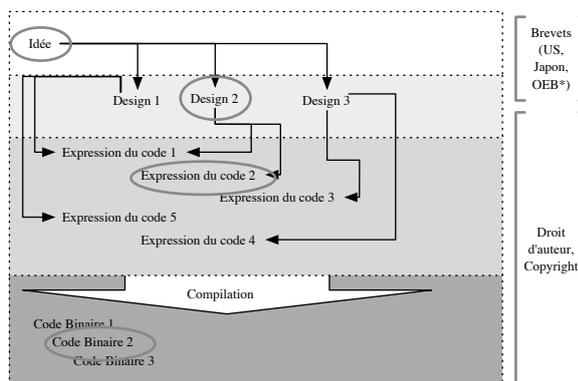


FIG. 2.2 – Développement non linéaire et Propriété Intellectuelle

2.1.2 Innovateurs et innovation, un intérêt convergent ?

L'apprehension de l'acte d'innovation dans l'industrie du logiciel est difficile. Pour autant, en se basant sur la dernière publication de la *Federal Trade Commission* aux Etats-Unis, un certain nombre de tendances peuvent être isolées :

- l'innovation dans cette industrie est de nature profondément cumulative,
- les innovations dans le domaine du logiciel requièrent considérablement moins de capitaux que dans d'autres industries High-Tech,
- la forme du changement technologique dans cette industrie est l'une des plus rapide qui soit,
- de nombreux schémas institutionnelles d'incitation à l'innovation existent (droit d'auteur, brevets, ...),
- cette industrie se trouve confrontée à l'introduction de la propriété industrielle non sans problèmes (brevets du logiciel)

Taxonomies de l'innovation dans l'industrie du logiciel

"Le chemin vers l'innovation est souvent de nature incrémentale, par ajout de nouvelles idées, et le développement et la commercialisation de nouveaux produits en utilisant les travaux anciens comme fondations et blocs de construction" *Microsoft Corporation, Dan Crouse, FTC*

La distinction entre les innovations radicales (illustrant, face à la continuité d'une technologie, une idée de profonde perturbation) et les innovations incrémentales (notion de profonde cumulativité, d'une dépendance historique et technologique de l'innovation nouvelle) est chose souvent difficile. Mais sont aussi opposées en général les innovations de produits aux innovations de procédés. Pour définir ces deux types d'innovations, le Manuel d'Oslo (2002, page 9) indique :

" on entend par innovation technologique de produit la mise au point/commercialisation d'un produit plus performant dans le but de fournir aux consommateurs des services objectivement nouveaux ou améliorés. Par innovation technologique de procédé, on entend la mise au point/adoption de méthodes de production ou de distribution nouvelles ou notablement améliorées. Elle peut faire intervenir des changements affectant ? séparément ou simultanément ? les matériels, les ressources humaines ou les méthodes de travail. "

Dans le domaine du logiciel, une innovation reste assimilée à une innovation de type service, de par sa nature intangible.

un exemple : La création des tableurs - de xxx à Excel (à développer)

Une question de capitaux

La nature physique du logiciel, et sa forme particulière de distribution potentielle (aisance de la copie, usage de l'Internet pour la diffusion, ...) réduisent d'autant la nécessité d'accumulation de capitaux pour le développement d'une application logicielle.

Technologies et évolutions

Puisque l'innovation est de nature incrémentale dans le domaine du développement logiciel, l'avance sur la concurrence est un moyen difficile pour conserver un avantage concurrentiel. Pour autant, peu d'industries peuvent se prévaloir, en matière de changement technologiques, à un tel dynamisme.

Institutionnalisation de l'acte d'innovation

Si la taylorisme fut en son temps cité comme une avancée majeure dans le domaine de la gestion organisationnelle de la production, il en est de même pour le système de *peer-review* offert par le modèle communautaire retrouvé dans le logiciel libre. Un logiciel est en majorité protégé par le droit d'auteur, et son pendant anglo-saxon, le copyright.

Et le brevet dans tout ça ?

L'introduction du brevet du logiciel peut représenter un challenge non négligeable pour une industrie habituée au droit d'auteur...

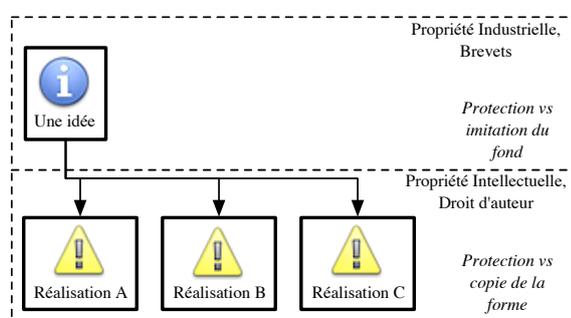


FIG. 2.3 – Licences et droits d'usages

Mais comment les logiciels sont-ils gérés et diffusés ?

2.1.3 Nature des licences et logiques de diffusion

Il va ici être abordé la problématique des licences, cherchant à démystifier les rapports entre logiciels propriétaires, domaine public, freeware, shareware, logiciels libres dits copyleft et logiciels libres "non copyleft". Les variables d'analyse retenues sont la "renonciation" à l'ensemble des droits par l'auteur les méthodes de distribution, les usages possibles par l'utilisateur final, la possibilité d'accès aux sources, et la nature des droits de modifications de celles-ci. Cette analyse est présentée dans le tableau suivant :

Un oeuvre dans le domaine public est une oeuvre abandonnée, dans le sens ou aucune obligation légales relatives au droit d'auteur ne s'y applique. Pour autant, un logiciel du domaine public n'inclut pas forcément son code source : il est possible que si seul l'exécutable ait

Finalité de la licence	"Renonciation" à l'ensemble des droits par l'auteur	Distribution, méthode	Usage, utilisateur final	Accès aux sources	Droit de modification des sources
Domaine Public	Oui	Libre, incontrôlable	Libre	Si disponible	Si disponible, libre
Freeware	Non	Pas de restriction	Libre	Non	Non
Shareware	Non	Selon Licence	Payant	Non	Non
"Propriétaire"	Non	Selon Licence	Payant	Non	Non
Logiciel Libre (Copyleft)	Non	Pas de restriction	Libre	Libre	Libre, "viral"
Logiciel Libre (Non Copyleft)	Non	Pas de restriction	Libre (parfois selon licence)	Libre	Libre, non "viral"

FIG. 2.4 – Licences et droits d'usages

été diffusé, passé le délai légal de protection, seul celui-ci tombe dans ce domaine. Shareware et freeware sont deux tournures de diffusion de codes binaires sous la forme d'exécutable. Un freeware est une application librement diffusée, à l'opposé d'un shareware, incluant en général un système d'essai avant achat. Ces deux finalités de licences d'utilisation restent propriétaires. Comme le nom l'indique, un logiciel propriétaire est un logiciel qui reste la propriété de son créateur : il n'en est vendu qu'une licence d'utilisation. Reste le Logiciel libre (l'appellation Open-Source regroupe à 95% des logiciels libres, les 5% restant regroupant les initiatives comme Shared Sources de Microsoft, ou autre, limitant l'accès aux sources à des conditions particulières - accords de confidentialité, ... -, voir interdisant la réutilisation des sources en question). Le logiciel libre ne sous-entend pas l'abandon du droit d'auteur (copyright, en anglais). Pour autant, il peut être fait usage des spécificités du droit d'auteur pour en permettre une certaine forme de diffusion (d'où l'expression copyleft, qui sous-entend, en cas d'usage d'un code diffusé sous licence copyleft, l'obligation de suivre de la licence de diffusion des nouveaux développements). Les fondements communs sont nombreux, dont le principal : l'accès au code source du logiciel. Mais une différence est fondamentale entre ces deux mouvements : le logiciel libre soutient une notion de liberté qui n'est pas au cœur de le logiciel libre. Ce sont donc les logiciels sous licences respectant les quatre libertés telles que définies par la Free Software Foundation : liberté 0 (liberté d'exécuter le logiciel pour n'importe quel usage) ; liberté 1 (liberté d'étudier le fonctionnement

d'un programme et de l'adapter à tous besoins) ; liberté 2 (liberté de redistribuer librement des copies) et liberté 3 (liberté d'améliorer le programme et de rendre publiques les modifications afin que l'ensemble de la communauté en bénéficie). Il est possible de reprendre rapidement l'analogie de la musique pour expliquer les raisons d'être de chacune de ces types de fonctionnalités de licence de diffusion : pour le domaine public, soit le délai légal de protection accordé par le droit d'auteur est dépassé (ce qui se retrouve pour de la musique classique), soit l'auteur livre son oeuvre au domaine public, et ce en présence ou non de la partition. Pour le logiciel propriétaire, l'auteur conserve ses partitions, et n'en distribue que des enregistrements. Gratuits dans le cadre du freeware, d'une durée ou d'une qualité limitée dans le cadre du shareware (avant achat), et payant dans le cas du propriétaire. Les licences open-sources incluent forcément la partition. Pour autant, l'aspect copyleft ou non marque une certaine forme (ou une absence) de restriction d'usage et de diffusion externe (la licence libre copyleft oblige, en cas d'usage de code sous licence libre copyleft, à diffuser le code nouveau sous la même forme). En majorité, la musique sera distribuée sous sa forme binaire, des compact disks à vendre. Mais l'artiste créateur souhaitant que le morceau de musique soit joué et diffusé non pas seulement par les quelques artistes ayant la capacité de recréer une codification du morceau, voit son intérêt résider alors à la diffusion de ses partitions : pour créer par exemple une oeuvre musicale non finie, une nouvelle tendance, ou tous ceux en ayant les capacités peuvent contribuer, et tous les audiophiles cherchant à s'initier, apprendre. L'artiste peut aussi par cette diffusion favoriser une logique de signalisation, de manière à se faire connaître. Tout en étant libre d'être parfois rétribué pour ses créations, mais plus souvent logiquement pour ses interprétations. Toutefois, un point reste fondamental : ce n'est pas parce que la partition est librement distribuée que l'auteur n'est pas cité comme auteur. Il reste donc associé à l'oeuvre, quelque soit sa diffusion. La logique n'a pas comme base seulement l'attribution de l'auteur. Par ce biais, il se fait ainsi connaître. Et par ce biais, outre les éléments de philosophie propre à une logique de partage, il peut vivre. Par contre, si le musicien n'est pas rétribué en temps qu'auteur-compositeur indépendant, mais est salarié, par exemple, d'une entreprise spécialisée dans l'organisation d'événements festifs. L'intérêt pour cette firme est d'obtenir le catalogue musical interprétable le plus large

possible permettant de répondre à toutes les attentes des clients. En matière de création musicale, elle peut avoir alors un intérêt en matière de productivité et de gestion des coûts de créer un catalogue de partitions partagées avec l'ensemble des concurrents. Un catalogue ouvert, créé de par un processus de coopération volontaire qui lui permettrait de réagir rapidement et de trouver une solution à toutes demandes. Parce que, quoi qu'il arrive, le jeu de la partition par l'artiste interprète en sera une adaptation personnalisée au goûts musicaux des clients. Autant d'arguments qui peuvent permettre de commencer à saisir l'intérêt à la coopération (sous sa forme la plus large) dans le domaine de la création.

2.1.4 Tendances

Tous types de licences confondus, un ensemble de tendances générales se dégage en matière de développement logiciel. En particulier :

- Une importance croissante en valeur et en volume des produits informatiques logiciels comparativement aux produits matériels (croissance d'usage de solutions de clusterisations logicielles par rapport aux investissements en super-calculateurs, ...).
- Une importance croissante des services liés aux logiciels au détriment de la production de logiciels à proprement parler (intégration de l'activité de conseil, ...).
- Une croissance notable du développement d'applications distribuées, le langage utilisé ou la plateforme cible perdant de l'importance de par l'usage d'interfaces compatible entre plusieurs composants logiciels distribués (usage des Web Services, ...).
- Une rationalisation notable des cycles de développement des logiciels (Usage de standards, architecture logicielle distribuée, ...).
- Un découpage des responsabilités entre aspects business et techniques (scindage des tâches en fonctions des rôles de chacun, ...).
- Une complexité croissante au sein des projets logiciels (intégration de composantes relatives à la sécurité, ...).
- Une orientation vers les standards de qualité (rationalisation, Démarche MDA : Model Driven Architecture, Normes ISO, ...).
- Une tendance à l'externalisation croissante des développements internes (rentabilité des investissements, ...).

Le développement informatique, et l'industrie du logiciel en particulier, est un secteur à la dynamique constante. Les évolutions des technologies y sont constantes. Cependant, la nature de ces évolutions est rarement à caractère radical, mais plutôt incrémentale. Par innovation radicale, il est entendu la notion de nouveauté absolue, que cela soit pour un produit ou une méthode. L'innovation incrémentale, pour sa part, se base sur les innovations passées pour apporter une nouveauté particulière. Notable, mais en continuité avec ce qui a été fait. En matière de propriété intellectuelle, le design est protégeable aux Etats-Unis par le Brevet et le droit d'auteur, alors que, dans l'attente du vote d'une directive autorisant le brevet du logiciel, seul le droit d'auteur s'applique en Europe (bien que des brevets du logiciel aient déjà été délivrés, dans l'illégalité, par l'Office Européen des Brevets). Une fois le code saisi, programmé, la protection qui s'applique alors relativement au produit logiciel relève du droit d'auteur et des droits voisins à celui-ci.

2.1.5 Logiciel, produit et cycle de vie

La représentation classique du cycle de vie d'un produit traduit les évolutions probables au cours du temps du nombre d'adopteurs d'un produit déterminé. Elle représente les différentes phases de l'historique commercial du produit, et est le plus souvent représentée sous la forme d'une courbe en cloche :

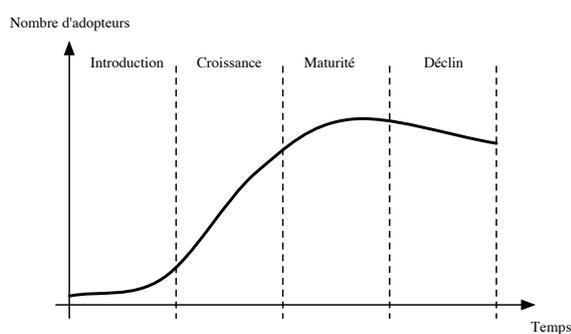


FIG. 2.5 – Cycle de vie d'un produit

Sont identifiés dans le cycle de vie du produit quatre phases distinctes. La phase d'introduction correspond à une période de faible croissance mais à fortes nécessités d'investissements, correspondant à la diffusion progressive du produit sur le marché. L'adoption

rapide du produit, révélateur de sa diffusion, correspond à la phase de croissance. La maturité marque le ralentissement des ventes, voir sa détérioration. Enfin, la période de déclin marque la fin du cycle de vie d'un produit, avec une érosion des utilisateurs. Cette représentation schématique est arbitraire. Pour autant, la logique sous-jacente peut s'appliquer pour tout type de bien, comme les logiciels par exemple. En effet, un logiciel commercialisé uniquement sous sa forme exécutable suit la même logique, le déclin marquant en général l'introduction d'une nouvelle version. La croissance d'un produit est liée à la capacité de commercialisation, de veille, ou encore de lecture des opportunités offertes par le marché. Une firme va ainsi chercher à maximiser son nombre de clients en fonction de ses capacités de démarchage. Ceci peut soulever un problème dans le cadre de technologies à caractère *incrémentales* dans leur adjonction d'innovations. Par exemple, les capacités disponibles au sein d'une firme peuvent empêcher l'avènement de certaines innovations. Soit parce que la firme ne dispose pas des ressources intellectuelles nécessaires, soit encore par manque de vision stratégique des décideurs. Une technologie peut être multiple dans ses usages et opportunités. Un entrepreneur va chercher à analyser et saisir toutes ces opportunités. Pour autant, il peut voir comme déclinante une technologie qui dont le nombre d'adopteur serait temporairement stagnant. Ce cycle de vie se retrouve dans les produits logiciels, quel qu'ils soient. La représentation en cloche présentée plus haut de ce cycle trouve une analogie directe avec les logiciels de type propriétaires. Le schéma suivant (figure 1.3) reprend les différentes étapes à la réalisation d'un projet de développement, en indiquant en proportion le temps passé en moyenne à chacune de ces étapes dans un projet de développement logiciel mené au sein du Centre de Recherche Public Henri Tudor.

La répartition de ce temps dans un développement logiciel moyen peut être estimée ainsi : l'analyse des besoins (règles de gestion, taxinomie des domaines, dans aspect statique et dynamique) représente à peu près 40% du temps total alloué au projet, le design 25%, le codage et l'intégration 20%. Enfin les 15% du temps restant sont alloués aux retro-actions entre tests (ceux effectués par les développeurs, et les tests fonctionnels réalisés par les utilisateurs) et debuggage du code, permettant la correction et l'optimisation du code développé. Le produit finalisé est alors obtenu, qu'il reste à déployer. En général, la maintenance sort

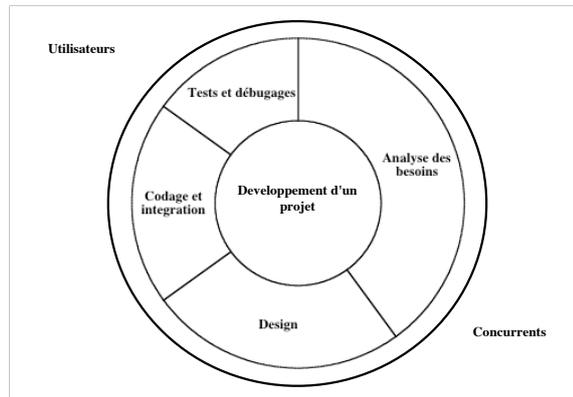


FIG. 2.6 – Cercle de développement, logiciel propriétaire

du cadre stricte de l'estimation liée au développement pour entrer dans un autre type d'activité (qui peut elle-même être sous-traitée). L'effet de retro-action des tests et débogages est important car un des principaux créateurs de connaissances nouvelles pour la firme ou l'équipe de développement. Comme représenté dans ce schéma, l'approche du projet est spécifique à son cadre de réalisation. Des utilisateurs peuvent participer à la réalisation du projet, mais pas sans restriction. La concurrence, de par la nature propriétaire du logiciel, est bien sûr exclue. Cela ne laisse que peu de possibilité pour la clientèle à agir sur le cycle de vie du produit en lui-même : une fois les besoins analysés, le processus de développement démarre. L'ajout de nouvelles fonctionnalités est difficile lorsque le projet arrive à sa fin.

2.1.6 Logiciel fermé, logiciel libre

Si il est possible d'assimiler un logiciel fermé (de type commercial ou non) à un bien classique, l'image est plus difficile à tenir dans le cas de logiciels libres. Il serait pourtant possible de parler de cycle de vie d'une plate-forme logiciel, de par la nature de la licence ouverte considérée. Il est ici question de logiciel libre, sans chercher à étendre cette analyse à l'ensemble des différents types de licences Open-Source. Ceci modifie d'autant le cercle de développement :

Pour autant que cela soit critiquable, le logiciel libre quitte la notion de produits pour entrer dans celui de technologies. De par la licence de diffusion et d'utilisation employée, une technologie perçue comme mature voir en déclin par certains utilisateurs peut rebondir

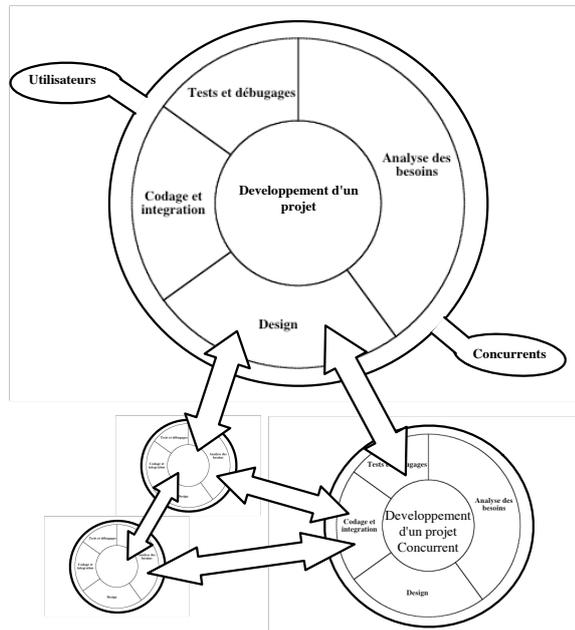


FIG. 2.7 – Cercle de développement, logiciel libre

en forte croissance si un tiers a la possibilité d'apporter, par exemple, son expertise d'un nouveau marché. Pour ce faire, l'accès au code source est fondamental.

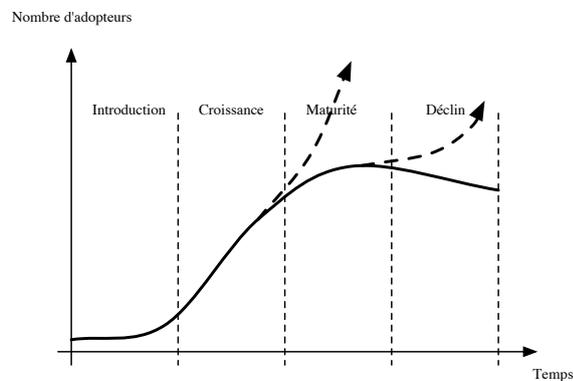


FIG. 2.8 – Cycle de vie du logiciel libre

Pour quelle raison ainsi différencier les cycles de vie ? Parce qu'un logiciel propriétaire est, comme son nom l'indique, propriété de son éditeur/propriétaire. Alors qu'un logiciel libre respectant les quatre libertés ne peut être assimilé à un produit unique puisque tout un chacun est libre de reprendre les sources pour les publier, s'il en a l'intérêt, sous un

autre nom. De plus, le droit de modification implique une possibilité de modification des sources mêmes, donc de la finalité du produit logiciel. Est-ce que cela peut marcher ? Un exemples simple illustrant la réussite d'une telle approche : Apache. Partit initialement des développements issus de la NCSA, ce serveur internet libre trouve son nom de l'acronyme a PATCHy server. Sa renommée n'est plus à faire (en mai 2003, 62% des serveurs http étaient des serveurs Apache, un chiffre illustrant le fait que cette solution est aujourd'hui la plus populaire sur Internet).

2.2 Caractérisation des utilisateurs

2.2.1 Pragmatiques et idéalistes

Deux grandes catégories de participants au mouvement open-source peuvent être isolées.

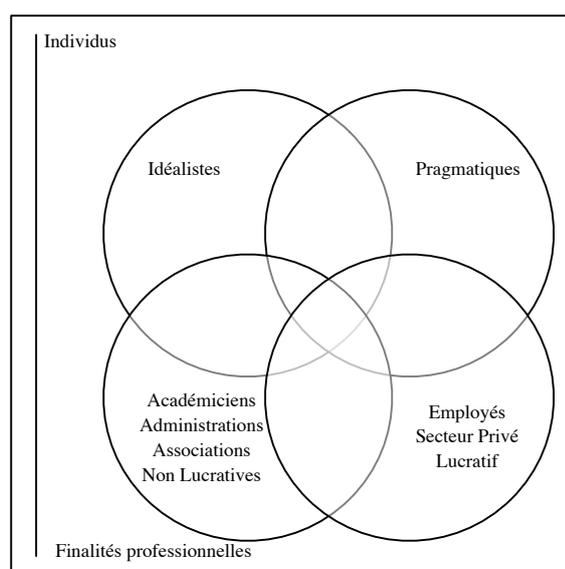


FIG. 2.9 – Populations et motivation

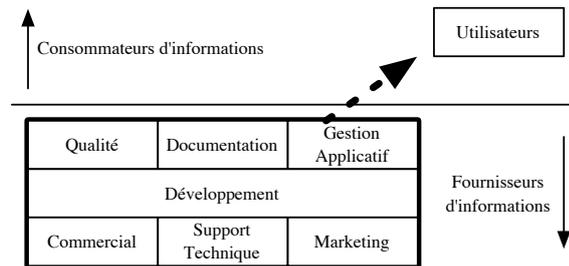


FIG. 2.10 – Communication mono-directionnelle

2.3 Axe Demande

2.3.1 Demande en logiciel propriétaire

2.3.2 Demande en logiciel libre

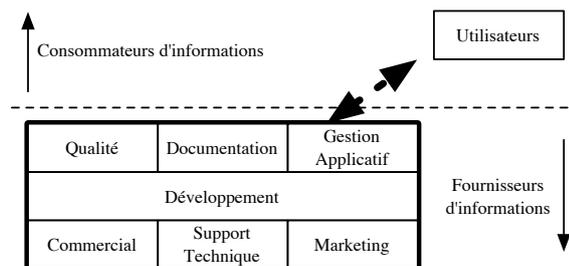


FIG. 2.11 – Communication bi-directionnelle

2.4 motivations

2.4.1 Administrations

Axe Offre

- création d'un bien commun : il est certains investissements qui peuvent être difficile à rentabiliser pour un investisseur privé, ou tout du moins soumis à des obligations quotidiennes de rentabilité. Si ce type d'investissements sont utiles et nécessaires pour l'ensemble de la société, l'Etat peut alors avoir intérêt à palier à ce défaut de marché. En matière de développement logiciel, la possibilité de création d'un bien commun, c'est

à dire utilisable par tous et qui plus est ne présentant aucune nuisance d'encombrement à l'usage peut représenter un investissement public de nature "idéale" : comme vu précédemment, une approche en terme de "produit" d'un logiciel libre est à proscrire. Son avantage principal face à un logiciel propriétaire est l'absence de toute perte face aux développements réalisés sous cette forme de licence. En effet, la mise à disposition des sources, et surtout la nature copyleft de la licence est un argument de poids quand à la possible pérennité des investissements réalisés.

- financement public : l'usage de moyens de financement issus de prélèvements publics directs ou non est en général soumis à une condition principale, à savoir l'assurance de la réalisation d'un bon investissement. Le logiciel libre, de par les différentes libertés d'usage et de réutilisation qui forment sa licence d'utilisation, présente la aussi un certain nombre d'avantages.
- transparence : le libre accès aux sources d'un logiciel permet la garantie de toute transparence quand aux produits logiciel développés.
- sécurité : "mille yeux ont plus de chance de trouver une erreur dans les sources d'un programme". Cet adage inspiré des travaux de E.S. Raymond est exact, à partir de l'instant où les capacités des développeurs sont équivalentes. Ceci permet à la fois d'observer une meilleure réactivité dans la production de patches et autre rustines comblant les trous de sécurités, mais aussi de s'octroyer les services d'une communauté ouverte aux capacités d'expertises certaines, et à coût réduit.
- interopérabilité :
- éducation

Axe Demande

- prix
- logique d'investissement
- Problème de la copie illicite

2.4.2 Firmes dont le développement est une activité primaire

Approche du libre autant idéaliste que pragmatique (?), car ce modèle de licence ne lui garantit pas des possibilités de situations de rente sur ses développements comme dans le cas d'une approche propriétaire. Pour autant, il existe de nombreux avantages à libérer son code source. Mais ces avantages seront différents selon la taille de la firme et la capacité d'impact sur les orientations stratégiques des standards liée à la variable taille.

Axe Offre

- Grandes entreprises : influencer sur les standards et les trajectoires de ceux-ci
- Grandes entreprises : rationaliser la plate-forme Unix (adieu Solaris, et autres, vive Linux)
- Petites et Moyennes Entreprises : accéder à de nouveaux marchés
- Tout type d'entreprises : baisse des coûts
- Tout type d'entreprises : accélération des cycles de développement

Axe Demande

- Baisse des coûts
- Problème de la copie illicite

2.4.3 Firmes dont le développement est une activité secondaire

Les firmes dont les activités de développement internes représentent une activité secondaire, par rapport à une activité principale tierce sont celles qui ont le plus à gagner. Il a été estimé que les emplois dans l'informatique et plus précisément dans le développement logiciel pour le secteur privé sont regroupés à 90% dans des firmes où l'activité de développement est une activité secondaire.

Axe Offre

- Rationalisation des investissements
- Baisse des coûts

Axe Demande

- Baisse des coûts
- Problème de la copie illicite

2.4.4 Individus**Axe Offre**

- Baisse des coûts

Axe Demande

- Baisse des coûts
- Problème de la copie illicite

2.5 Innovation et succès

Xerox aura reconnu l'importance stratégique de l'innovation. En 1970, cette entreprise américaine créa le Palo Alto Research Center dans le but de développer de nouvelles idées. Depuis sa création, ce parc fut à l'origine de nombreuses innovations. Historiquement, la liste est longue des nouveautés, prototypes et créations qui en sont issus. Dans le courant des années 70, il y fut par exemple développé le "Alto", un ancêtre de l'ordinateur personnel comprenant une souris, une interface homme-machine graphique, et un éditeur de texte avancé. De plus, le Parc fut aussi à l'origine d'avancées capitales dans le domaine des réseaux informatiques et de l'impression numérique laser. Toutes ces innovations devaient obtenir une reconnaissance du marché. Mais problématiquement, aucune de celles-ci ne se retrouvèrent dans des produits Xerox. En effet, ces inventions et innovations devaient générer la fortune de plusieurs sociétés tierces comme Apple, Microsoft, IBM, ...

Pour quelle raison ? Le développement d'une innovation n'est qu'une partie du travail. En effet, Xerox n'avait alors pas développé ses capacités en termes organisationnels permettant l'accompagnement de l'innovation vers sa commercialisation. Ainsi, Xerox tenta de vendre son ordinateur, l'Alto, en utilisant les ressources et réseaux de vente de ses produits "main-frame", le Xerox Computer Services. Ce fut un échec retentissant : XCS n'avait pas les

moyens de comprendre les spécificité de ce nouveau marché. Liée à des routines comportementales, cette division ne sût s'adapter à ses nouvelles exigences.

L'autre problème est lié à la maturité du marché. Trop tôt pour Xerox, qui, après cet échec, se recentra vers d'autres activités, laissant à Apple le soin de récupérer les résultats de leur R&D en matière d'interface utilisateurs pour développer l'ancêtre du Macintosh, le Lisa. A nouveau, erreur stratégique pour Apple, qui aura put imposer son système d'exploitation Mac OS sur le marché pour peu que celui-ci fut plus ouvert... laissant du même coup à Microsoft la possibilité de devenir leader sur le marché des systèmes d'exploitation grand publics.

Ce rapport entre l'innovation et l'imitation est ainsi complexe. Etre le premier n'est pas forcément un avantage, car le marché peut ne pas être mature, ou la cible de consommation visée fausse.

Cette histoire est véridique. Il y a un grand perdant, Xerox, un perdant potentiel, Apple, et un grand gagnant, Microsoft.

Cette histoire s'est aussi réalisée à une époque où la propriété intellectuelle ne s'était pas "radicalisée" comme à l'heure actuelle... Si Xerox avait pu breveter son interface utilisateur, comme c'est maintenant le cas, l'histoire de l'informatique aurait été irrémédiablement changée.

Chapitre 3

Modèles d'exploitation du logiciel libre

reprendre les différentes questions - Cf. Christophe Feltus.

3.1 Entreprises

3.1.1 Utilisation

3.1.2 Développement

3.2 Citoyens

3.2.1 Utilisation

3.2.2 Développement

3.3 Exemples existants de modèles d'exploitation

3.3.1 Externaliser un logiciel

Cisco et ses imprimantes

3.3.2 Partager le développement

HP et Samba

3.3.3 Produire des produits connexes

O'Reilly

Perl

3.3.4 Mode dualiste : libre et propriétaire

Sleepycat Software

Dansguardian

3.3.5 Mode dualiste séparé : libre et propriétaire

Redhat

3.3.6 Mode services libres + logiciels propriétaires

3.3.7 Mode libre - fondation

à mettre dans les produits connexes ?

Chapitre 4

Application des théories économique

Partie de Matthieu à ajouter.

Chapitre 5

Le cadre légale

Introduction à définir.

5.1 Licences logicielles Libres

L'application des 4 libertés (décrite dans le chapitre d'introduction 1) est réalisée dans le cadre d'utilisation du droit d'auteur via des licences logicielles décrivent ces libertés sous la forme d'un "contrat"¹. Ces licences libres sont les bases légales de l'existence du logiciel libre mais aussi de sa pérennité et des moyens de distribution. Elles sont classées dans deux grands types, les licences libres dites "copyleft" et les licences libres dites "non-copyleft". Cette nuance est importante d'un point de vue légal mais surtout dans les cadres de distribution des logiciels libres.

Pour qu'une licence soit qualifiée de libre, il faut qu'elle respecte les 4 libertés du logiciel libre?. Une liste ²de ces licences libres est maintenue par la Free Software Foundation. Il existe aussi une autre qualification des licences libres via l'Open Source Initiative ³(OSI). La grande majorité (plus de 95%) des licences qualifiées par l'OSI sont des licences libres car l'Open Source définition ⁴ définit les 4 libertés en d'autres termes qui se rejoignent. La différence est une subtilité pour quelques licences logicielles qui sont peu utilisées dans la communauté du logiciel libre.

Une licence logicielle utilise le droit d'auteur. Ce droit exclusif de l'auteur lui permet de

¹La terminologie exacte est une licence logicielle ou un contrat de licence logicielle. Il existe des différences subtiles entre les deux décritent dans une autre section

²<http://www.gnu.org/licenses/license-list.fr.html>

³<http://www.opensource.org/>

⁴<http://www.opensource.org/docs/definition.php>

définir ses droits par rapport à l'oeuvre littéraire. En effet, les programmes d'ordinateur sont assimilés comme oeuvres littéraires et artistiques (au sens de la convention de Berne⁵). Souvent l'utilisation du droit d'auteur, dans les programmes d'ordinateur, est de concéder très peu de droits aux utilisateurs comme le fait le logiciel propriétaire. Par contre, dans le cadre des licences libres, on concède plus de droits/libertés aux utilisateurs. Le droit d'auteur donne à l'auteur toutes les possibilités de libérer son oeuvre ou une partie sous des conditions permissives. L'auteur de logiciel libre donne des droits sur son oeuvre comme la distribution, la modification et la distribution de ces modifications. Dans ce cadre des licences libres, l'auteur peut ajouter une condition sur la protection des droits accordés. Il décide que l'oeuvre soit libre sans protection de cette liberté, c'est à dire qu'elle peut rester ou même devenir propriétaire. Ou il décide que l'oeuvre soit libre avec la protection de cette liberté, c'est à dire que l'oeuvre est disponible mais doit toujours rester libre. Voici la différence principale entre les oeuvres libres "non-copyleft" et "copyleft", cette différence fondamentale permet d'avoir deux approches assez différentes du cadre de distribution des logiciels libres.

— debut à supprimer

Il existe plusieurs licences libres, les plus connues sont la GNU General Public License (ref), la licence de type "Berkeley" (ref) ou la Mozilla Public License (ref).

— fin à supprimer

5.1.1 Les licences libres "copyleft"

Le type "copyleft" (en français : la gauche d'auteur ⁶) est largement utilisé par la communauté du logiciel libre. Par exemple, la GNU General Public License (la licence la plus largement utilisée avec près de 60% de part) est une licence de type copyleft. Ce qui permet aux auteurs et aux utilisateurs de savoir que le logiciel restera toujours libre et ainsi de conserver des droits équitables envers tous.

⁵<http://www.wipo.int/clea/docs/fr/wo/wo001fr.htm>

⁶La gauche d'auteur est un simple "jeu de mots" sur le droit d'auteur. Bien entendu, les licences libres utilisent le droit d'auteur et ce "jeu de mots" n'est que montrer l'utilisation alternative de ce droit

Exemples de licences libres “copyleft”

- GNU General Public License
- GNU GNU Lesser General Public License
- Affero General Public License

5.1.2 Les licences libres “non-copyleft”

Le type “non-copyleft” est aussi largement utilisé par la communauté. Par exemple, les systèmes d’exploitation BSD⁷ sont distribués sous des licences de ce type. Ces licences permettent une certaine flexibilité car les logiciels libres sous ce type de licence peuvent devenir des logiciels propriétaires. C’est pour cela qu’un nombre important de constructeurs utilisent dans des produits propriétaires, des logiciels libres sous des licences libres “non-copyleft”.

Exemples de licences libres “non-copyleft”

- MIT X11 (version originale)
- *BSD-like License

5.1.3 Les licences non-libres

Il existe plusieurs licences qui ressemblent à des licences libres mais c’est une fausse impression. En effet, plusieurs fausses licences libres existent et il est important de pouvoir les identifier pour éviter des conflits et/ou les problèmes légaux.

5.1.4 Comment appliquer une licence libre ?

Le droit d’auteur est automatique. Si vous créez une oeuvre originale, vous avez automatiquement le droit d’auteur qui s’applique. Il ne faut pas faire d’enregistrement particulier ou autre, il s’applique par défaut. Bien entendu, il est important d’avoir des informations précises sur le régime de droits d’auteur.

⁷FreeBSD,NetBSD,OpenBSD

Un exemple pratique : La société Baz s.a. désire distribuer un logiciel libre sous la licence GNU General Public License. Une procédure existe à la fin de la licence elle-même pour l'appliquer correctement. Comme la licence est elle-même assez volumineuse, pour chaque fichier source, la société appliquera une clause par défaut dans chaque fichier qui informe d'une part du copyright daté de la société ainsi que les informations sur la licence elle-même.

Un exemple de clause copyright avec l'application d'une licence libre pour une oeuvre de la société Baz s.a.

Copyright (C) 2002,2003,2004 Baz s.a. Luxembourg This program is free software ; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation ; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY ; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program ; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

Ensuite, la société Baz s.a. devra inclure la licence dans le paquet contenant le logiciel (souvent sous le nom "LICENSE" dans la racine du paquet).

L'application même de la licence n'est pas compliqué mais il faut prendre en compte les points connexes comme les contrats de travail, les méthodes de distribution mais aussi les assignements de "copyright".

5.2 Licences libres et les obligations contractuelles

5.3 Licences libres non-logiciels

5.4 Logiciel libre et brevet

Chapitre 6

Conclusion

Conclusion commune (m.farcot et adulau) à définir.

Annexe A

GNU Free Documentation License

GNU Free Documentation License
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free

software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding

them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word

processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License to the Document are reproduced in all copies, and that you add no conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute Opaque copy (directly or through your agents or retailers) of edition to the public.

It is requested, but not required, that you contact the authors of the

Document well before redistributing any large number of copies, to them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you the Modified Version under precisely this License, with the Version filling the role of the Document, thus licensing and modification of the Modified Version to whoever possesses a of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History of the Document). You may use the same title as a previous if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of Document (all of its principal authors, if it has fewer than unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors,

and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, list them all as Invariant Sections of your combined work in license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name different contents, make the title of each such section unique adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled " and any sections Entitled "Dedications". You must delete all sections Entitled

"Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and it individually under this License, provided you insert a copy of License into the extracted document, and follow this License in other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works When the Document is included in an aggregate, this License does apply to the other works in the aggregate which are not derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may translations of some or all Invariant Sections in addition to original versions of these Invariant Sections. You may include translation of this License, and all the license notices in Document, and any Warranty Disclaimers, provided that you also the original English version of this License and the original of those notices and disclaimers. In case of a disagreement the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and automatically terminate your rights under this License. parties who have received copies, or rights, from you under License will not have their licenses terminated so long as parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions

will be similar in spirit to the present version, but differ in detail to address new problems or concerns. <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of License "or any later version" applies to it, you have the option following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit situation.

If your document contains nontrivial examples of program code, we

recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public to permit their use in free software.

Biographical Note

Cette section devrait contenir des notes biographiques.