

Table des matières

Introduction.....	2
1.Définition.....	2
2.Historique.....	2
3.Principes.....	3
4.Licences.....	3
4.1.Domaine public.....	3
4.2.Licences de type BSD.....	4
4.3.Licences de type GPL.....	4
5.Communautés.....	4
6.Les fondations.....	5
7.Le modèle économique.....	5
8.Les grands projets open source.....	6
9.Les Outils de développement open source.....	6
10.Avantages et limitations de l'open source.....	7
Conclusion.....	8
Annexe.....	9

introduction

Les logiciels open sources commencent à devenir une alternative sérieuse pour les logiciels propriétaires. En effet un logiciel open source mûr propose la majorité des fonctionnalités de son concurrent propriétaire pour un prix nul ou largement inférieur. Ce succès étant dû à une approche non classique de développement, nous allons présenter dans le présent rapport les caractéristiques de base de l'approche Open Source ainsi que ses aspects communautaires et économiques.

1. Définition

Un logiciel open source est un logiciel qui est fourni avec l'autorisation pour quiconque de l'utiliser, de le copier, et de le distribuer, soit sous une forme conforme à l'original, soit avec des modifications, ou encore gratuitement ou contre un certain montant. Ceci signifie en particulier que son code source doit être disponible.

Selon les différentes communautés et entreprises, le terme logiciel open source relève une ambiguïté avec le terme logiciel libre, notamment en ce qui concerne sa gratuité. En pratique on confond les deux notions, mais le mieux reste de se référer à la licence du produit en question pour identifier s'il est gratuit ou non.

2. Historique

Les dates clés de l'histoire de l'open source sont:

- *Fin des années 70* : Richard Stallman, informaticien la MIT, se révolte contre les logiciels propriétaires.
- *1983*: Stallman lance le projet GNU qui a pour objectif de construire un UNIX dont la totalité des logiciels est librement partageable.
- *1985*: Stallman quitte MIT pour fonder la Free Software Foundation.
- *1991*: Linus Torvalds, un étudiant finlandais de l'université de Helsinki, s'appuie sur GNU pour concevoir LINUX.

3. Principes

La Free Software Foundation maintient une définition du logiciel libre basée sur quatre libertés:

1. **Liberté 1** : La liberté d'exécuter le programme, pour tous les usages
2. **Liberté 2** : La liberté d'étudier le fonctionnement du programme. Ceci suppose l'accès au code source.
3. **Liberté 3** : La liberté de redistribuer des copies. Ceci comprend la liberté de vendre des copies.
4. **Liberté 4** : La liberté d'améliorer le programme et de publier ses améliorations. Ceci encourage d'une part la création d'une communauté de développeurs améliorant le logiciel , et permet d'autre part le *fork*, soit la création d'une branche de développement concurrente, notamment en cas de désaccord entre développeurs.

De part ces libertés, les utilisateurs, les développeurs et les entreprises jouissent des mêmes droits que le propriétaire du programme, excepté son droit de propriété.

4. Licences

À l'exception des logiciels dans le domaine public, les logiciels libres sont protégés comme tout logiciel par le droit d'auteur. La particularité des logiciels libres est que l'auteur renonce à l'exclusivité de la plupart des droits que lui donne le droit d'auteur. Il distribue le logiciel accompagné d'une licence libre qui énumère les droits donnés à l'utilisateur.

Les logiciels libres sont distribués avec de simples licences. Généralement, ils sont également distribués sans la moindre garantie.

Certaines licences, dont la plus connue et utilisée pour les logiciels libres, la licence publique générale GNU, sont relativement complexes. Ainsi la GPL ne donne le droit de redistribuer un logiciel que si l'ensemble du logiciel, y compris toutes les éventuelles modifications, sont redistribués selon les termes exacts de la GPL. Cette licence est dite *virale* ou *contaminante*, car si elle autorise la fusion d'un logiciel sous GPL avec un logiciel sous une autre licence, elle n'autorise en revanche la redistribution de la fusion que sous GPL.

Les licences des logiciels libres sont souvent divisés en trois, selon le degré de liberté accordé par la licence en matière de redistribution.

4.1. Domaine public

Il ne s'agit pas réellement d'une licence, mais simplement du fait que le logiciel n'a aucun droit. Chacun est libre de faire ce qu'il veut avec. Théoriquement, tout logiciel tombe dans le domaine

public une fois les droits d'auteur échus.

4.2. Licences de type BSD

Il s'agit des licences qui offrent la plus grande liberté. En général, seule la citation des auteurs originaux est demandée. En particulier, ces licences permettent de redistribuer un logiciel libre sous une forme non libre. Ces licences permettent donc à tout acteur de changer la licence sous laquelle le logiciel est distribué. Un cas de changement de licence courant est l'intégration de logiciel sous licence BSD dans un logiciel sous *copyleft* (licence GPL). Un autre cas courant est l'intégration de logiciel sous licence BSD dans les logiciels propriétaires.

Ces licences sont notamment utilisées par la Berkeley Software Distribution (licence BSD) et Apache Software Foundation (licence Apache).

4.3. Licences de type GPL

Il s'agit des licences qui obligent la redistribution sous une licence libre. Les licences du projet GNU sont les plus célèbres. Une telle licence permet d'intégrer du logiciel sous licence BSD et de le redistribuer sous licence GPL, alors que l'inverse est impossible.

Le degré de liberté moindre des licences de type *copyleft* a été critiqué par des acteurs des projets BSD et des acteurs commerciaux; Ce qui a poussé la Free Software Foundation à proposer la LGPL. La différence entre GPL et LGPL réside principalement dans le fait que la LGPL permet de lier un programme tiers non GPL à une bibliothèque LGPL, sans pour autant révoquer la licence. Ainsi, il devient possible à un programmeur désireux de faire un logiciel propriétaire d'utiliser certains outils du libre (ex: la bibliothèque graphique GTK).

5. Communautés

Pour qu'un projet open source puisse survivre face à la concurrence, il a besoin d'une communauté pour l'adopter. Celle-ci est constituée de bénévoles, d'étudiants, de chercheurs et des utilisateurs. Certains de ces contributeurs ont même évolué pour former des sociétés.

Ces communautés s'organisent autour d'un patron physique ou moral qui joue le rôle de l'administrateur du projet. Le *core team*, de sa part, se charge de contrôler la qualité des contributions soumises par tous les participants.

Les membres des communautés se servent des outils bien spécifiques afin de collaborer entre eux, ces outils sont basés en général sur la connexion internet. Les plus utilisées sont les forums des sites web spécialisés comme les incontournables SourceForge, Gforge et freshmeat. On peut ajouter à

cela les newsgroups, les CVS, le Bugtraq et le canal IRC.

6. Les fondations

Les communautés open source ont besoin d'un cadre qui va les organiser et les allier sur une même voie. D'où la nécessité d'organisme qui vont standardiser les principes et définir les licences d'utilisation.

La première de ces organisations est la *Free Software Foundation* (FSF). C'est une organisation américaine à but non lucratif, fondée par Richard Stallman en 1985 pour aider au financement du projet GNU et de la communauté du logiciel libre. Elle a fondé la première version de la licence GPL.

Ensuite elle vient L'*Open Source Initiative* (OSI) qui est une organisation dédiée à la promotion des logiciels libre et atteste de la validité des licences Open Source.

Une autre organisations est L'*Open Source Development Labs* (OSDL). C'est une organisation à but non lucratif créée par un consortium d'entreprise (IBM, HP, Intel, Nec, CA) dont la mission consiste à accélérer les développements autour des logiciels libres mais aussi de les simplifier et d'en organiser la cohérence.

Enfin, les fondations qui gèrent la vie d'un projet. Il existe, parmi les plus connues, la fondation Apache et la fondation Debian. Ce ne sont pas des sociétés, comme Red Hat ou Novell, mais des associations qui dirigent et coordonnent les développements pour leur domaine d'activité.

7. Le modèle économique

Le modèle économique des SSSL (sociétés de services en logiciels libres) est lié principalement à la notion de service : vendre un savoir-faire et une expertise plutôt qu'un droit d'usage sur un logiciel. Ce modèle consiste à facturer au client non pas le logiciel lui même, qui reste open source, mais le service de support, d'installation, de personnalisation, de configuration et de maintenance que l'on vend autour sous forme d'abonnement.

Un autre système payant, utilisé par d'illustres éditeurs de logiciels libres comme *MySQL*, consiste en une licence commerciale qu'achètent certains clients afin d'intégrer le produit libre dans leurs propres produits dans le but de les redistribuer de manière payante, avec un code source fermé. Cette stratégie assure par exemple les deux tiers des revenus de *MySQL*.

Une troisième forme de financement pour les éditeurs de logiciels libres est l'organisation de séminaires pour la présentation des solutions open sources au grand public, notamment aux commerciaux.

Le modèle économique des logiciels open sources a suscité toutefois certaines critiques:

- incompatibilité entre les besoins des clients et les demandes de la communauté.
- Les revenus sont générés par le service, et non pas par le logiciel lui même.
- Difficulté du maintien sur le long terme d'une ressource humaine compétente et capable de tenir des délais compatibles avec la notion de service.

8. Les grands projets open source

Distributions

- *Red Hat* : Société commerciale éditrice de logiciels libres, notamment connue pour sa distribution GNU/Linux.

Système et interface

- *KDE* : Le bureau virtuel le plus courant sous Linux.

Navigateurs

- *Firefox* : Issu du projet Mozilla , Firefox est désormais le concurrent de Internet Explorer.

Sécurité

- *Snort* : Logiciel de détection d'intrusion sur les réseaux IP (IDS).
Il permet de sniffer le réseau, générer des logs d'événements et de détecter les intrusions.

Bureautique

- *OpenOffice* : Suite bureautique libre, concurrente de Microsoft Office.

9. Les Outils de développement open source

Développement

- *Eclipse* : Environnement intégré Open source de développement. C'est une plate-forme commune sur laquelle viennent se greffer des plugins, ce qui le rend configurable par tous les éditeurs.
- *NetBeans* : Environnement open source de développement. Ecrit en Java, il supporte n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre NetBeans. C'est un produit gratuit, sans aucune restriction quant à son usage.
- *Hibernate* : Framework Java Open Source utilisé pour assurer la persistance des données. C'est un outil de mapping objet/relationnel assurant la correspondance entre les tables relationnelles et les classes Java.
- *Ant* : Langage et interpréteur qui effectue les tâches de compilation d'applications et d'exécution de procédure, d'une manière analogue aux .bat de MS/DOS ou à Make sous

Unix.

Développement Web

- *Struts* : Framework de développement d'applications Web en Java du projet Apache Jakarta. Il combine les technologies JSP et Servlets suivant le modèle MVC2.

Bases de données

- *MySQL* : Version Open Source et gratuite de SQL.

Serveurs Web

- *Apache* : Logiciel qui permet d'animer un serveur WEB HTTP. C'est le plus utilisé au monde.
- *Tomcat* : Serveur HTTP, compatible J2EE, qui implémente les technologies Java Server Page et servlet java. Les applications réalisées pour Tomcat peuvent être portées sur des serveurs d'applications d'entreprise sous une quelconque plateforme. Il peut être intégré aux serveurs Apache et IIS.

10. Avantages et limitations de l'open source

10.1. Avantages

a) Des logiciels de qualité

un programmeur qui expose son code au monde et aux autres participants du projet fera en sorte que le code soit de qualité. Si ce n'est pas le cas, un autre programmeur du projet proposera à la place un code plus efficace ou plus propre. De plus, un grand nombre de personnes participent en permanence aux tests des logiciels libres, ce qui permet d'identifier les bugs très tôt.

b) Un coût raisonnable pour les utilisateurs

les logiciels Open Source sont soit **gratuits** soit proposés à un coût raisonnable, dans tous les cas inférieur à celui des logiciels commerciaux équivalents

Si des bugs sont identifiés, la prise en compte de la correction de ces erreurs est assurée gratuitement par les participants du projet.

c) Une forte réactivité

On constate en général que les utilisateurs ayant un besoin précis d'amélioration du logiciel

reçoivent un écho favorable et rapide de la part de la communauté des développeurs du projet. Un utilisateur peut lui même coder une fonctionnalité dont il a besoin, en accord avec le coordonnateur du projet.

d) Des logiciels à long terme

les projets aboutis de logiciels libres ont une très longue durée de vie, car il y a toujours des volontaires pour poursuivre le projet.

10.2. limitations

a) Les obstacles financiers

Les capitaux risqueurs ne se précipitent pas pour financer des projets d'éditeurs de solutions Open Source. De plus les façons de génération de profits restent assez restreintes en quantité et en qualité.

b) Orientation vers les aspects techniques

La plupart des projets Open Source réussis concernent des "logiciels techniques", c'est-à-dire des logiciels non pas applicatifs mais remplissant une fonctionnalité technique utilisée par d'autres logiciels. Les leaders des logiciels applicatifs d'entreprise que sont Microsoft (bureautique) ou SAP (ERP) n'ont en effet aucunement été détrônés par les logiciels Open Source comme Linux ou Compiere.

c) Le support et la formation des utilisateurs

Beaucoup d'utilisateurs de logiciels ont besoin non seulement d'un support technique pointu pour corriger les bugs éventuels, mais surtout d'un support de base pour être aidé dans l'utilisation du logiciel, voire de formations.

d) Le marketing et l'inertie de marché

Les premiers projets Open Source ne disposaient pas de budgets marketing. Cependant, on voit apparaître de plus en plus de projets Open Source appuyés ou financés par des éditeurs de logiciels ou des intégrateurs.

conclusion

Malgré les contraintes financières et juridiques qui limitent jusqu'à présent une véritable croissance des logiciels open sources, ceux-ci restent plus prometteuses que les produits propriétaires, surtout pour les petites sociétés et les pays en cours de développement (comme le Maroc). Reste à atteindre un niveau de maturité suffisant pour passer des solutions techniques vers des solutions d'entreprises (les ERP pas exemple) qui puissent rivaliser dans le marché international.

Annexe

Héritaire : propriété d'un type de licence. Les développements réalisés à partir d'un logiciel sous ce type de licence doivent être diffusés sous cette même licence.

Contaminant : propriété d'un type de licence. Lorsqu'un développement quelconque est intégré à un autre logiciel sous ce type de licence, le résultat doit être diffusé sous cette licence.

copyleft : principe général imposant que les versions modifiées d'un programme libre doivent aussi être des logiciels libres.

Shareware : c'est un logiciel qu'on peut légitimement se procurer gratuitement, mais qu'on doit payer si l'on désire l'utiliser.

Freeware : on utilise le terme freeware pour les logiciels propriétaires qui sont distribués gratuitement.

Communauté : La communauté des logiciels libres désigne des personnes qui participent à des projets, qu'ils soient les développeurs ou les utilisateurs du produit résultant.

Domaine public: un logiciel du domaine public est un logiciel non soumis aux droits d'auteurs.

Logiciel propriétaire: c'est un logiciel dont l'utilisation, la redistribution ou la modification sont interdites ou exigent une autorisation spécifique.

Logiciel commercial : Le logiciel développé par une entreprise dans le but de gagner de l'argent sur l'utilisation de ce logiciel. Un logiciel libre peut être commercial.

Bazar : connotation pour la communauté open source

Cathédrale : façon dont les logiciels propriétaires ferment le code de leurs applications.