3TC37: Contribution à un logiciel libre

Modèles économiques

Marc Jeanmougin
Théo Zimmermann

Télécom Paris, Institut Polytechnique de Paris

L’open source à la conquête des entreprises

The Cathedral and the Bazaar

Essai d'Eric Raymond (1997) comparant deux modèles de développement :

- La **cathédrale** : modèle de développement traditionnel, "vertical".

  Exemples : Emacs, gcc (à cette époque)…

- Le **bazar** : modèle de développement flexible, "horizontal" (aujourd'hui on dirait "agile").

  Un principe : "Release early, release often".

  Et une loi : "Given enough eyeballs, all bugs are shallow".

  Exemples : Linux, fetchmail (expérience d'application du modèle).

Mozilla

By Mozilla Corporation, MPL 2.0

- Netscape : navigateur dominant entre 1995 et 1997, quand Internet Explorer écrase le marché en étant **distribué avec Windows**.
- En 1999, l'entreprise **Netscape libère le code de son navigateur** qui devient Mozilla (puis Mozilla Firefox).
- Firefox parvient à être un **concurrent sérieux d'Internet Explorer** jusqu'à ce que Google Chrome arrive et écrase tout à son tour.
- Création de la **Mozilla Foundation** en 2003.
- À l'origine de la **licence MPL 2.0**.

StarOffice, OpenOffice, LibreOffice

By Christoph Noack, CC BY-SA 3.0

- Star Division, une entreprise allemande publiant la suite StarOffice, est rachetée en 1999 par Sun Microsystems.
- En 2000, **Sun Microsystems libère le code** de la suite qui devient OpenOffice.
- Création en 2005 du format standard OpenDocument.
- **Oracle rachète Sun Microsystems** en 2009.
- The Document Foundation est créée en 2010 pour créer un **fork communautaire** : LibreOffice.

Android

By Google, CC BY-SA 3.0

- Système d'exploitation open source **basé sur Linux**, développé par Google et publié à partir de 2007.
- Permet de **concurrencer Apple** et l'iPhone (plus de 70% du marché).
- N'est pas développé de manière collaborative ou ouverte, mais la licence **autorise les fabricants** de smartphones à en produire leur **version dérivée**.
- En pratique, la plupart des smartphones sous Android sont **loin d'être libres** car ils incluent aussi les Google Mobile Services (Google Play) qui sont propriétaires.

Microsoft et l'open source

- Open Letter to Hobbyists (1976) : Bill Gates se plaint du "**vol**" de ses logiciels.
- Steve Balmer compare Linux à "un **cancer** qui s'attache à tout ce qu'il touche" (2001).
- Microsoft commence à contribuer à Linux en 2009, publie des langages de programmation open source (F# en 2005, **TypeScript** en 2012).
- Satya Nadella écrit "Microsoft <3 Linux" en 2014 (Microsoft fait tourner Linux dans Azure depuis 2012), Microsoft devient l'un des plus gros **sponsor de la Linux Foundation** en 2018.
- **VS Code** est publié en open source en 2015.
- Le **Windows Subsystem for Linux** est introduit en 2016.
- Microsoft rachète **GitHub** en 2018.

Le logiciel libre a-t-il gagné ?

- On pourrait croire que oui si on regarde les **outils utilisés par les développeurs** sur leurs ordinateurs (éditeurs, gestionnaires de version, compilateurs, bibliothèques).
- Mais si on regarde en termes de **libertés des utilisateurs finaux**, on en est loin !
- La plupart des applications web et mobiles **dépendent de logiciels libres** mais ne donnent pas de libertés à l'utilisateur.
- GitHub n'est pas libre ! (Même si des alternatives libres, comme GitLab, Gitea, Forgejo, etc. existent.)
- Et Microsoft Word continue aussi à dominer le marché face à LibreOffice.

Modèles économiques

Le modèle classique

- Dans le monde des logiciels propriétaires :
  Revenus basés sur la **vente de licences** d'utilisation.
- Production de logiciels: coût élevé de développement, coût de la copie très bas (quasiment nul).
- C'est un modèle de *rente* qui peut être extrêmement profitable (dépend davantage du nombre de clients que des efforts fournis par le vendeur).
- Modèle similaire pour la musique, les films, etc.

L'exclusion d'usage commercial

- Logiciel **gratuit pour les usages non commerciaux** (ou pour les très petites entreprises / indépendants) mais payant pour les usages commerciaux (ou au-delà d'une certaine échelle).
- Cela permet d'avoir une plus large communauté d'utilisateurs (voire de contributeurs) sans renoncer à l'économie de rente.
- Même lorsque le code est public, ce n'est **pas du logiciel libre**, et beaucoup d'utilisateurs ou contributeurs potentiels en auront conscience. Par exemple :
  - **Pas de garantie sur l'avenir du logiciel** en cas de faillite / renoncement / rachat de son producteur.
- Exemples actuels : SublimeText (éditeur), Obsidian (notes).

Modèles économiques du logiciel libre

1. Les modèles basés sur le *développement* payant et non la *vente*.
2. Les modèles basés sur la vente de versions propriétaires étendues par rapport à une version libre et gratuite (*open core*) ou bien de licences spécifiques.
3. Les modèles qui ne sont pas basés sur la vente des logiciels mais sur des services annexes : maintenance, hébergement, etc.

Chaque acteur peut combiner plusieurs modèles à sa guise. Dans la suite, on se focalise sur des *éléments* constitutifs de modèles économiques.

Développement payant

- Un client a besoin d'un certain produit logiciel.
- Il peut embaucher un développeur, ou une entreprise (éditeur de logiciel) qui développe le logiciel.
- Le client (qui est *a priori* propriétaire du logiciel qu'il a fait développer à ses frais) peut choisir de publier le produit sous licence libre pour des raisons diverses.
- Modèle fréquent lors de la commande de logiciels par des institutions publiques (notamment aux US).

Vente d'extensions (modèle *open core*)

- Deux versions :
  - Une version libre.
  - Une version propriétaire **étendue avec plus de fonctionnalités** (qui peuvent finir par être ajoutées à la version libre).
- Les entreprises sont encouragées à acheter la version étendue :
  - Choix d'un vocabulaire (trompeur) :
    "Community Edition" / "Enterprise Edition"
  - **Vente combinée** de services (déploiement/maintenance) avec les extensions.
- Avantages :
  - Si la version libre n'est pas sous licence copyleft, alors le recours à un CLA / CTA n'est pas nécessaire.
  - Les distributions Linux peuvent inclure la version libre, qui sera néanmoins à jour.
- Exemple : Modèle passé de Netscape (licence MPL), actuel de GitLab (licence MIT).

Vente d'exceptions

- Publication sous licence libre *copyleft* (par exemple GPL).
- Vente de licences qui permettent l'utilisation du logiciel dans des produits qui ne seront pas publiés sous GPL.
- C'est un modèle que la FSF trouve acceptable, contrairement au précédent (mais qui requiert un CLA ou CTA) : https://www.gnu.org/philosophy/selling-exceptions.html
- Exemple :

  

  un des systèmes de bases de données les plus importants (entreprise rachetée par Sun pour un milliard de dollars, et ensuite par Oracle).

Vente de maintenance / support : exemple de RedHat

- Fondé en 1994.
- Chiffre d'affaires : 1 milliard de $ en 2012.
- Résultat net : 199 millions de $ en 2012.
- Nombre d'employés : 6 100 en 2014.
- Vente de support : entre 350$ et 2 500$ par an et serveur.
- Basée sur une distribution gratuite et libre : *Fedora*
- 82% des revenus (2012) viennent de la vente de contrat de support, le reste de la formation et du conseil.
- 18% des coûts (2012) sont pour le développement Open Source (les résultats reviennent donc à la communauté).

Hébergement (modèle SaaS) : exemple de Zulip

- SaaS = Software as a Service
- Zulip est un logiciel libre de chat / alternative à Slack très populaire depuis quelques années.
- L'entreprise Zulip propose d'héberger des serveurs pré-configurés et automatiquement mis à jour :
  - Gratuitement pour les projets open source, les équipes de recherche…
  - De manière payante pour les entreprises.

Financement par les dons

Désormais communément admis que les projets open source ont besoin d'être soutenus financièrement.

Quelques succès célèbres comme celui d'Evan You, le créateur de Vue.js.

Mais, d'après Overney et al. (2020) :

- Beaucoup de projets demandent des dons pour supporter les activités de développement.
- Mais **rares sont ceux qui reçoivent suffisamment** pour financer un développeur à plein temps. Par exemple sur npm :
  - Seuls 2 paquets sur 1000 réclament des dons.
  - Seuls 5% de ceux qui en réclament reçoivent > 1000$/mois.
- Beaucoup de projets (sur Open Collective) ne dépensent pas l'argent collecté (thésaurisation) ou le dépensent pour d'autres utilisations (exemple : frais d'hébergement).

Financement ponctuels par des bourses / crowdfunding …

- Bourses de fondations :

  Exemple : NLNet
  - une fondation soutenant l'internet ouvert,
  - aujourd'hui distribue à + de 90% de l'argent de l'UE (**Next Generation Internet** initiative),
  - a financé des dizaines de projets open source (sur des missions ponctuelles et bien définies)

- **Financement participatif** (crowdfunding) : une campagne de dons ponctuelle pour mobiliser la communauté des utilisateurs afin de financer une amélioration spécifiques.

- Autres : missions diverses, bounties, etc.

Exercice

Quel est le modèle de financement de Firefox ? GIMP ? Krita ? Blender ? Gitlab ? Linux ?