Logiciel Libre ou Open Source : de l'idéologie à l'influence

1. Logiciels et licences 2
   La pluralité du concept de "logiciel" 2
   Code source et fichier exécutable 2
   Protection juridique du logiciel 3
   Diffusion et licence d'utilisation 5
2. Du Logiciel Libre à l'Open Source 6
   Les quatre libertés du logiciel libre 7
   Licences permissives et émergence de l'Open Source 8
3. Microsoft, du « cancer Linux » à la rémission Open Source 9
   Opposition du logiciel libre 10
   La transition vers l'Open Source 11
Conclusion 11
Webographie 13

1. Logiciels et licences

La grande majorité des objets utilisés au quotidien comporte une part d'informatique, c'est à dire une association de matériel électronique (le hardware) et de composants logiciels (les firmware et software). Si certains circuits électroniques se passent toujours de composant logiciel, ne réagissant qu'aux flux électriques les traversant pour effectuer une séquence d'opérations logiques, la miniaturisation des processeurs fait que tout composant électronique peut intégrer une certaine complexité reposant sur un programme informatique. Ainsi, une télécommande, une machine à laver basique ou un portail électrique contiennent une partie logicielle.

La pluralité du concept de "logiciel"

Le logiciel est omniprésent et n'est pas cantonné aux matériels munis d'un écran ou aux objets dits connectés : l'entrée du numérique dans le secteur automobile est bien antérieure à la révolution Tesla, les dizaines de capteurs présents dans une voiture reposent sur des composants logiciels depuis des décennies[1]. L'avènement de l'Internet des Objets (en anglais *IoT - Internet of Things*), qui voit l'intégration de capacités informatiques complexes dans un nombre d'objets du quotidien grandissant - dans l'électroménager notamment - rend particulièrement visible cette omniprésence du logiciel qui apporte de l'“intelligence” - et surtout des capacités de communication - dans des objets qui en étaient auparavant dénués. Mais le logiciel, c'est avant tout ce qui rend utile l'ordinateur - et ses succédanés que sont *smartphone*/ordiphone et tablettes tactiles -, sa raison d'être : l'ordinateur n'est que l'architecture matérielle nécessaire au fonctionnement du logiciel qui, par sa programmation, va exécuter les tâches pour lesquelles l'utilisateur le sollicite - qu'elles soient liées à la productivité ou au divertissement.

Enfin, un logiciel ne fonctionne que rarement en autonomie, mais plutôt dans un environnement complexe. Un navigateur web, un logiciel de bureautique ou un jeu vidéo s'appuient non seulement sur le système d'exploitation de la machine qui le fait fonctionner - qu'il soit une variante de Windows, de Linux, de MacOS ou d'Android, pour ne citer que les plus courants - mais également sur une multitude de logiciels annexes. Si un programme informatique est un assemblage de lignes de codes rassemblées dans une multitude de fichiers, tout nouveau logiciel n'est pas écrit en partant d'une page blanche : il s'appuiera par exemple sur des bibliothèques, d'autres programmes déjà écrits avec des fonctionnalités connues qui seront intégrées. Il s'appuiera également sur des programmes externes, des services fonctionnant soit sur la même machine, soit sur d'autres, sur le même réseau local ou accessible via Internet.

Code source et fichier exécutable

Pour pouvoir être exécuté par une machine, le code qui décrit le programme doit subir une opération, la compilation, qui va être la traduction du code source intelligible pour et écrit par l'humain en un code optimisé exécutable par la machine. Ce code devient un fichier binaire incompréhensible pour un humain, et effectuer l'opération inverse - la décompilation -, est une opération complexe ne permettant pas de retrouver le code source original, bien qu'il

[1] À titre d'exemple, les systèmes d'ABS - pour éviter le blocage des roues lors du freinage - contrôlés par microprocesseur font leur apparition entre la fin des années 60 et le début des années 70

puisse permettre de comprendre le fonctionnement du programme. Code source et fichier exécutable sont les deux faces d'un même programme informatique.
Même s'il est lisible par un humain, le code source demande certaines connaissances préalables pour pouvoir être compris. Sa lecture, tout comme la lecture de n'importe quel écrit, implique de connaître le langage dans lequel il a été produit. De plus, il est l'ensemble des instructions permettant au programme de fonctionner. Son organisation - son architecture -, les algorithmes qui sont utilisés, la façon dont sont traitées les données qui transitent par le programme : tout est contenu dans le code source. Il est le fruit du travail de l'intelligence humaine qui a œuvré à résoudre le problème qui lui était posé. Sa production a eu un coût, toujours en temps, souvent en salaire également ; il représente un investissement, sur lequel un retour est donc attendu. Le point suivant traitera donc de la protection de cet investissement.

Protection juridique du logiciel

Comme mentionné dans l'introduction, le code source d'un logiciel est protégé par le droit d'auteur. La définition du terme "Logiciel" est donné dans un arrêté du 22 décembre 1981 :

> Logiciel (n. m.) : Ensemble des programmes, procédés et règles, et éventuellement de la documentation, relatifs au fonctionnement d'un ensemble de traitement de données (en anglais : *software*).

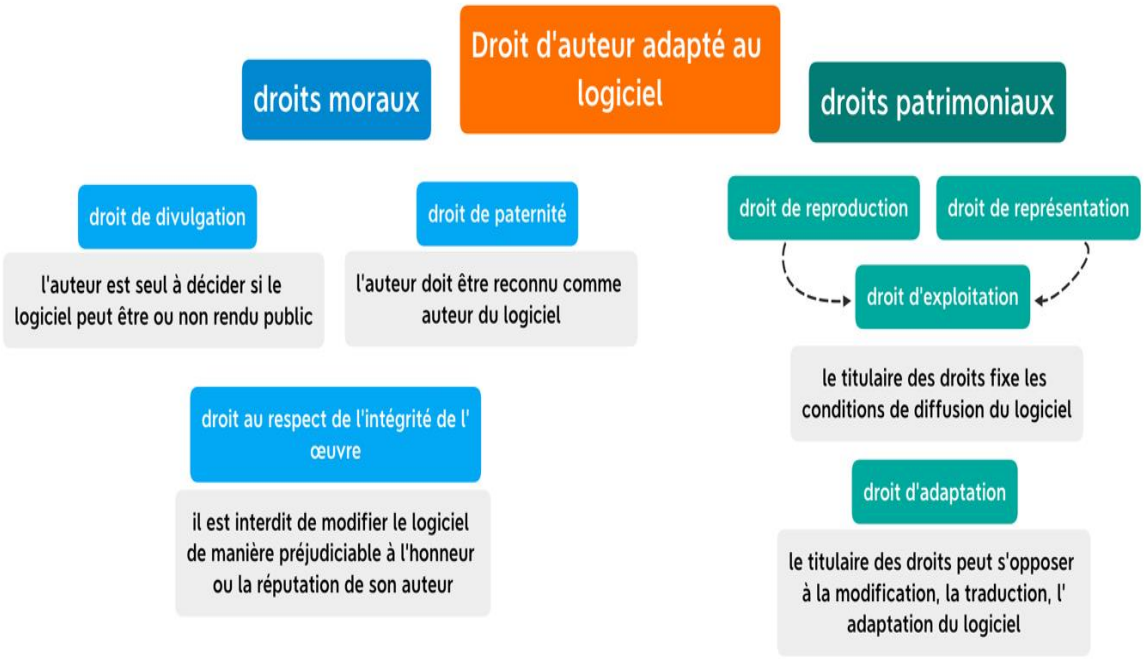
En France, c'est la Loi n° 85-660 du 3 juillet 1985 qui intègre dans le logiciel comme œuvre de l'esprit aux yeux de la loi[2], en modification de la Loi n° 57-297 du 11 mars 1957 sur la propriété littéraire et artistique. Abrogée par la Loi n° 92-597 du 1er juillet 1992 et codifiée dans le Code de la propriété intellectuel, elle définit des droits moraux, incessibles, et des droits patrimoniaux. À noter que si dans la version initiale de loi, « le logiciel créé par un ou plusieurs employés dans l'exercice de leurs fonctions appartient à l'employeur auquel sont dévolus tous les droits reconnus aux auteurs », la Loi n° 94-361 du 10 mai 1994 précise que ce sont « les droits patrimoniaux sur les logiciels et leur documentation » qui sont dévolus à l'employeur et non pas l'intégralité des droits. L'application du droit d'auteur est automatique et ne nécessite aucune action pour bénéficier de sa protection.

Aux États-Unis, la notion de propriété intellectuelle qui est associée n'est pas le droit d'auteur mais le copyright - donc littéralement le droit de copie. Contrairement au droit latin, la Common Law n'intègre pas de droits moraux dans les sujets de propriété intellectuelle : l'œuvre n'est pas considérée comme consubstantielle à son auteur. Il s'agit uniquement de protéger l'investissement qui a permis la création de l'œuvre, les coûts liés à sa reproduction et sa diffusion étant considérés comme comparativement marginaux : sans protection, n'importe qui pourrait en tirer bénéfice sans avoir eu à supporter les coûts liés à sa création. Ce point sera repris explicitement dans le rapport de la *National Commission on New Technology Uses of Copyrighted Works* (CONTU), publié le 31 juillet 1978 :

> « les ordinateurs devenant moins encombrants, moins chers et plus puissants, ils se retrouvent chez les particuliers et sur leur lieu de travail quand, dans le même temps, les programmes informatiques sont de moins en moins dédiés à un type de machine

[2] Article 1er, V.

spécifique. [...] Si le coût de duplication d'une information est bas, alors il est plus simple pour une personne peu scrupuleuse de la dupliquer ».



L'apparition du micro-ordinateur a en effet fait passer l'équipement informatique du statut de machine-outil dédiée à une tâche spécifique à celui d'instrument flexible capable d'adresser des problématiques sortant du cadre purement professionnel.

Le rapport de la CONTU est un travail visant à compléter la *Revision of the U.S. Copyright Act* de 1976 qui intègre la notion de "programme informatique" comme œuvre soumise à la protection du *copyright* ; cette révision rend également la protection automatique par *copyright* automatique, alors qu'auparavant il était nécessaire d'effectuer une démarche de publication formelle, au risque de voir l'œuvre considérée dans le domaine public le cas échéant. À partir de l'entrée en vigueur du texte, la mention d'un copyright sur une œuvre assure sa protection.

Le rapport précise que « le coût de développement d'un programme informatique est bien plus élevé que son coût de duplication » et identifie 4 profils patrimoniaux de créateur de programme informatique[3] :

1. Celui qui « couvre ses coûts et fait un profit suffisant lors de la première vente, peu concerné par les publications ultérieures ».
2. Celui qui « réparti ses coûts en vendant de multiples copies intégrant un système de protection contre les copies non autorisées ».
3. Celui dont les « coûts de développement sont pris en charge par, par exemple, le gouvernement ou une fondation ».
4. Celui pour qui « le coût n'est pas un problème et offre son travail au domaine public ».

[3] ibid., p. 11

Ces 4 profils peuvent être rapproché de modèles d'affaire qui ont accompagné le développement de l'industrie informatique :

- Le premier profil peut être associé au modèle des Entreprises de Service Numérique[4], qui peuvent être amenées à développer une solution répondant aux besoins spécifiques d'un client, et qui sera facturée à un prix fort en tant que création sur mesure - même si aujourd'hui l'abondance de l'offre logicielle fait que le travail sera plutôt fait autour de l'adaptation et l'intégration de solutions existantes.
- Le second correspond au modèle standard des éditeurs de logiciels présents sur le marché : des logiciels grand public - jeux vidéo, applications mobiles - aux solutions professionnelles - modélisation 3D, logiciels métiers -, en passant par les suites bureautiques qui font le lien entre univers professionnel et personnel.
- Les troisième et quatrième profils vont se rapprocher des développements fait par des équipes informatiques interne à des structures de tout type - par exemple, universitaires - qui, avec la démocratisation des compétences en développement informatique vont créer des solutions présentant un intérêt en dehors de leur organisation.

Diffusion et licence d'utilisation

La question de la protection du logiciel a été liée à la démocratisation du micro-ordinateur. Tant que la diffusion des programmes informatiques était cantonnée à un milieu fermé, le risque de voir se disséminer un programme de manière incontrôlée était restreint, surtout que la compatibilité des programmes avec différents matériels était très limitée. Ainsi, les logiciels étaient généralement distribués avec leur code source, autant pour des aspects de documentation que de maintenance : en cas de problème, il était ainsi possible de chercher la cause et la fixer.

L'accès au grand public à l'informatique, liée à la plus grande accessibilité du matériel tient autant aux évolutions technologiques qui en ont réduit l'encombrement et le prix qu'à l'arrivée de programmes simplifiant la manipulation par les utilisateurs. En baissant les prérequis nécessaires à l'utilisation d'un ordinateur, les logiciels grand publics faisaient entrer l'informatique dans le marché de masse. Or, fournir les codes sources des logiciels en même temps que les fichiers exécutables, c'était diffuser largement le mode de fonctionnement interne du logiciel et s'ouvrir à la copie. Pour les sociétés dont la vente de logiciels était la source de revenu principale, laisser la possibilité de copier et modifier le fonctionnement du logiciel était un non-sens économique, d'autant plus que le code source n'est pas nécessaire au fonctionnement des fichiers exécutables. De plus, en vertu de la protection intellectuelle associée au logiciel, l'obtention une copie d'un logiciel s'accompagne de conditions d'utilisation, contractualisées au travers d'une licence dont l'acceptation est un prérequis à l'utilisation du logiciel.

En complément du fait que la distribution d'un fichier exécutable empêche l'utilisateur de prendre connaissance de la mécanique de fonctionnement interne du logiciel, une des dispositions les plus courantes dans les licences d'utilisation du logiciel est l'interdiction de la

[4] ESN, anciennement SSII pour Société de Service en Ingénierie Informatique

rétro-ingénierie[5]. Cette pratique - qui n'est pas réservée au logiciel[6] -, consiste à chercher à comprendre les mécanismes internes de fonctionnement d'un produit, par tous les moyens possibles. Dans le cas du logiciel, il peut s'agir de la décompilation du fichier exécutable, mais également d'une étude de toutes les fonctionnalités pour comprendre comment elles interagissent entre elles : quel impact à la modification de paramètres de configuration, comment se comporte le logiciel en fonction des données entrées par l'utilisateur, quelles sont les interactions entre le logiciel et son écosystème (le système d'exploitation, les autres logiciels installés, les services accessibles par le réseau...). La finalité peut aussi bien être de comprendre les raisons d'un *bug* que de chercher à modifier le comportement, effectuer une copie du logiciel ou exploiter une faille pour accéder à des données protégées.

La licence définit donc quels sont les droits accordés à l'utilisateur dans le cadre de l'utilisation du logiciel : sur combien de machines le logiciel peut être installé, combien d'utilisateurs sont autorisés à en bénéficier, si l'utilisateur à le droit de revendre le logiciel... Posant des restrictions fortes, ce type de fonctionnement par licence très fermée donna naissance au mouvement du Logiciel Libre, qui sera l'objet de la partie suivante.

2. Du Logiciel Libre à l'Open Source

Le mouvement du Logiciel Libre vient se positionner en opposition au modèle du logiciel dit propriétaire, qui restreint les libertés de l'utilisateur. Le principe de logiciel "propriétaire" ne prend existence qu'à partir de la naissance du logiciel "libre", qui vient s'y opposer. La doctrine libre parle d'ailleurs plutôt de logiciel "privateur", pour accentuer le fait que les logiciels non-libres empêche leurs utilisateurs de bénéficier de toute la liberté dont ils devraient jouir dans l'usage qu'ils en ont. Ainsi, si la notion de "privateur" est très claire, la notion de "propriétaire" doit bien évidemment être prise dans un sens proudhonien de propriété, c'est à dire du vol.

L'idée que l'utilisateur devrait pouvoir être libre d'utiliser de la manière la plus complète possible les logiciels dont il a l'usage est intrinsèquement liée à l'essor de l'informatique. Durant les 20 premières années du développement de l'informatique moderne, le public utilisant des ordinateurs étaient principalement des personnes techniques, passionnées, capables de comprendre le fonctionnement d'un programme informatique et d'aller le modifier. L'accès au code source était donc une évidence pour ces gens puisqu'ils étaient en mesure d'intervenir sur ledit code au besoin. Le mouvement du Logiciel Libre est ainsi issu du mouvement *Hacker*, dont le sens est bidouilleur : face à un problème, le hackeur va aller creuser dans les tréfonds techniques du problème pour construire une solution, la finalité qu'est la résolution du problème étant plus importante que l'“élégance” de la méthode employée, l'important ensuite étant de pouvoir partager la solution avec le reste de sa communauté. Cette notion de partage chère à la communauté Hacker tient autant à une volonté d'élévation de l'autre qu'à la fierté de résoudre des problèmes techniques complexes.

[5] Aussi appelée ingénierie inverse, de l'anglais *reverse-engineering*.

[6] Cette pratique existe dans toutes les industries : analyse chimique en parfumerie ou dans l'agroalimentaire, démontage et analyse des différents composants mécaniques et électroniques dans l'automobile, l'aéronautique ou la défense...

L'exemple érigé en légende fondatrice du mouvement du Logiciel Libre a lieu en 1980, au AI Lab du MIT[7]. Richard M. Stallman, programmeur brillant travaillant sur place, fait face à un problème de bourrage papier avec le nouveau système d'impression du laboratoire. S'il avait réussi à contourner les problèmes récurrents de l'ancienne imprimante en bidouillant son programme, il lui était impossible de reproduire une solution équivalente : le fournisseur refusait de lui fournir les codes sources du nouveau serveur d'impression. Il lui était impossible de régler rapidement le problème parce que le constructeur lui refusait la liberté de le faire.

Les quatre libertés du logiciel libre

L'épisode de l'imprimante n'est qu'un des événements qui amèneront Stallman à fonder, en 1985, la Free Software Foundation[8]. Son objectif est de favoriser le développement des logiciels libres, dans l'intérêt des utilisateurs. Le concept de logiciel libre s'articule autour de 4 principes, appelés les 4 libertés et notées de 0 à 4 - parce qu'en informatique, le premier élément est généralement noté l'élément 0

La liberté 0 vient répondre aux limitations classiques d'usage couramment prévues dans les licences logicielles telles que :

- La restriction du nombre de personnes autorisées à utiliser une licence, ou du nombre de machines sur lequel peut être déployé le logiciel (ce qui oblige à acheter un nombre de licences correspondant à une certaine volumétrie).
- La restriction des usages, certains logiciels proposent différentes versions à différents prix, soit avec des fonctions limitées, soit avec des destinations d'usage limitées (interdiction d'utilisation pour effectuer une activité commerciale, par exemple, qui nécessite l'obtention d'une licence plus chère).

La liberté 1 nécessite l'accès au code source, et nécessite des connaissances techniques pour pouvoir l'exercer. La liberté 0 est essentielle à la liberté 1, puisque c'est uniquement si l'usage est libre que la possibilité de modifier le programme a du sens.

La redistribution entendue dans la liberté 2 n'est pas nécessairement gratuite. Une des confusions les plus courantes est de penser que le libre est forcément gratuit - accentué par

[7] Laboratoire d'Intelligence Artificielle du Massachusetts Institue of Technology
[8] *Free* est à traduire par libre, jamais par gratuit : "*Think of free as in free speech, not as in free beer*"

les 2 sens de *free*. Il n'est pas interdit de vendre des copies d'un logiciel libre, même si le logiciel a été acquis gratuitement.

La liberté 3 est celle qui a le plus d'impact, puisqu'elle signifie que l'auteur original d'un logiciel n'a pas de contrôle sur l'évolution de celui-ci. Elle implique de partager le logiciel, mais également son code source, pour que d'autres utilisateurs puissent à leur tour le modifier. Cette liberté a une portée très forte car, si le système du *copyright* ne fait que peu de cas du droit moral, le respect de l'intégrité de l'œuvre et de la paternité sont des notions essentielles dans le système du droit d'auteur.

Pour permettre l'application de ces grands principes en accord avec le droit, Stallman publie en 1989 la première version de la licence publique générale GNU[9]. Son préambule indique que « la licence accordée par la plupart des éditeurs de logiciels tente de garder les utilisateurs sous le contrôle desdits éditeurs. En contraste, notre licence publique générale GNU cherche à assurer votre liberté de partager et de modifier le logiciel libre -- de garantir que le logiciel restera libre pour tous ses utilisateurs. ». Une notion forte mise en place avec la GNU GPL est la notion de *copyleft*, en opposition au *copyright*. Le *copyleft* est une utilisation du *copyright*, c'est à dire l'utilisation du mécanisme légal qui régit la propriété intellectuelle des logiciels dans la *common law*, pour garantir qu'un logiciel créé libre le restera. Le principe du *copyleft* est que la liberté 3 (la distribution du code modifié) n'est pas un droit, mais un devoir : si un logiciel libre est modifié par utilisateur et redistribué ensuite, il doit obligatoirement être distribué avec ses sources et l'autorisation d'être à son tour modifié. C'est l'obligation de réciprocité, l'aspect le plus sensible du logiciel libre.

Licences permissives et émergence de l'Open Source

Si le *copyleft* permet de garantir qu'un logiciel créé libre restera libre - c'est à dire qu'il répondra aux quatre libertés fondamentales -, il est aussi un frein à l'adoption des logiciels libres. En effet, si beaucoup de logiciels libres sont à destination des utilisateurs finaux non informaticiens, une très grande partie de la production libre consiste en des outils destinés à être intégrés dans des projets de plus grande envergure : ce qu'on appelle des bibliothèques logicielles, mentionnées en amont dans ce rapport. Or, bien qu'il soit possible d'intégrer une bibliothèque libre dans un projet non-libre - en mentionnant son utilisation -, s'il est nécessaire de modifier ladite bibliothèque pour qu'elle soit parfaitement adaptée aux besoins du projet, alors il sera nécessaire de publier lesdites modifications effectuées. Le risque étant à ce moment-là, si le cloisonnement du code n'est pas fait correctement - c'est à dire que le code libre modifié se retrouve mélangé avec du code qui n'était pas destiné à être publié -, est la publication de code confidentiel. En parallèle et à la suite de la GNU GPL sont apparues d'autres licences, dont certaines dites permissives - au premier rang desquelles les licences BSD[10] et MIT[11] -, c'est à dire n'impliquant pas d'obligation de réciprocité, tout en respectant les 4 libertés fondamentales : la production de licence n'étant pas une fin en soi mais plutôt un moyen de garantir l'aspect libre d'un logiciel, c'est l'évolution du mouvement du logiciel libre qui a fait grossir la liste des licences existantes.

[9] GNU General Public License, ou GNU GPL, ou plus simplement GPL. GNU fait référence au projet GNU, un système d'exploitation libre initié par Stallman.
[10] Berkeley Software Distribution License, de l'Université de Berkeley, Californie
[11] MIT License, du Massachusetts Institute of Technology

Le projet libre le plus connu est bien évidemment le système d'exploitation GNU/Linux, apparu au début des années 90 et s'appuyant en partie sur des logiciels libres issus du projet GNU démarré en 1983 par Stallman. Implémentation sous licence libre d'un système d'exploitation reprenant le fonctionnement des systèmes Unix équipant les universités, GNU/Linux a servi de socle à la démocratisation du logiciel libre. En effet, GNU/Linux a apporté un environnement de développement s'appuyant sur la philosophie de liberté et de partage de la connaissance portée par le mouvement libre.

Cependant, le succès dont bénéficie GNU/Linux, lié au développement commercial d'Internet - les deux révolutions s'alimentent mutuellement - n'est que moyennement compatible avec les exigences du logiciel libre et son positionnement politique trop marqué, peu versé dans le compromis. C'est pourquoi au milieu des années 90 apparaît aux côtés du côté du *free software* la notion d'*open source*, qui amène à la création de l'Open Source Initative en 1998 sous l'impulsion d'Eric S. Raymond.

Si le mouvement du Logiciel Libre se concentre sur les libertés absolues qui doivent être accordées aux utilisateurs par le logiciel, l'*Open Source Definition* se focalise sur ce qui caractérise une licence *Open Source*. La différence est subtile mais réelle puisqu'elle oppose l'idéologie libre face à un certain pragmatisme, notamment économique : le logiciel libre milite pour la liberté des utilisateurs avec une revendication morale, l'*open source* pour l'accès au code source des logiciels avec une motivation économique. La vision portée par le mouvement Open Source met en avant les bénéfices de travailler avec des codes sources ouverts : une immense communauté liée par des expertises techniques visant l'excellence pour les solutions sur lesquelles elle travaille.

Cette vision est associée à des méthodes de travail qui auront un impact profond sur le métier du développement logiciel : apparition des méthodologies agiles, suivi de l'engouement pour les méthodes *Lean*[12], dont le point d'orgue sera la publication en 2008 the *The Lean Startup* d'Eric Ries.

3. Microsoft, du « cancer Linux » à la rémission Open Source

S'il existe une société qui permet de témoigner de l'évolution du secteur du logiciel au cours des 40 dernières années, il ne peut s'agir que de Microsoft. Fondée en 1975 par Bill Gates et Paul Allen, le 5e des GAFAM[13] a une position particulière dans l'écosystème du logiciel autant par l'évolution de son positionnement que par l'influence de son célèbre fondateur.

Peu après avoir arrêté ses études à Harvard pour créer Microsoft, Bill Gates fait publier une lettre ouverte dans la *newsletter* du *Homebrew Computer Club*, rassemblement de passionnés de l'informatique personnelle naissante dans la Silicon Valley, pour plaider contre le piratage des logiciels. En effet, ayant passé un contrat avec un vendeur de matériel informatique pour

[12] Méthodes de travail inspirées par le Toyota Production System, sur lequel s'appuiera la marque pour relancer son activité au sortir de la guerre. Théorisées dans les années 90 par James P. Womack, Daniel Roos et Daniel T. Jones, elles mettent au cœur du système de production les notions d'amélioration continue des processus, de la qualité et de la satisfaction client.

[13] Acronyme désignant les 5 géants technologiques américains que sont Google, Amazon, Facebook, Apple et Microsoft

distribuer, avec versement de *royalties*, un logiciel propriété de Microsoft, il se rend compte que des versions piratées de son logiciel sont en circulation. Son message est clair : « Ce que vous faites [en volant des logiciels], c'est empêcher de bons logiciels d'être écrits. Qui peut se permettre de fournir un travail professionnel pour rien ? »[14]. Cette lettre sera publiée en début d'année 1976, année de la révision du *U.S. Copyright Act*.

Les choix stratégiques de Microsoft et les contrats noués avec IBM qui définissent un standard de micro-ordinateur avec l'IBM-PC, pour lequel Microsoft fourni un système d'exploitation tout en gardant le droit de fournir un équivalent pour les clones d'IBM-PC, assurent le succès de Microsoft. MS-DOS puis Windows sont les systèmes d'exploitation présents par défaut sur les ordinateurs personnels, sans compter les logiciels de ce qui deviendra la suite bureautique Office.

Pour défendre ses intérêts, Microsoft rejoint la *Software Publishers Association* (SPA) en 1986, association de lutte contre le piratage des logiciels, avant de fonder la *Business Software Association* (*BSA*) en 1988. Le rôle de ces associations de lutte contre le piratage est d'aller auditer les parcs informatiques des entreprises pour vérifier que les logiciels installés disposent de licences en règle ; ce sont aussi des structures de lobbying défendant les intérêts des éditeurs de logiciel. Microsoft quittera ensuite la SPA lorsque cette dernière se portera partie civile aux côtés du gouvernement U.S. pour son abus de position dominante.

Opposition du logiciel libre

Si dès la création de Microsoft, Bill Gates s'est trouvé en opposition avec les tenants d'une diffusion libre des logiciels, l'émergence d'Internet et de GNU/Linux a exacerbé les tensions entre la firme multinationale qui ne tarderait pas à être poursuivie par le gouvernement américain au titre de la loi antitrust et les communautés libres qui désignent Microsoft comme étant le grand Satan. Le bras droit de Gates et nouveau CEO, Steve Ballmer, rend à la communauté libre la politesse en expliquant que si « le mouvement Open Source est une concurrence intéressante qui force Microsoft à innover, Linux est un cancer qui s'attache à tout ce qu'il touche en matière de propriété intellectuelle. C'est comme ça que fonctionne sa licence » [15].

La concurrence de l'*Open Source* commence en effet à prendre de l'ampleur avec Internet qui s'est largement démocratisé. Or, contrairement au monde du micro-ordinateur personnel que Windows domine largement, le monde des serveurs constituant la toile du Web est principalement propulsé par les logiciels open source. Le problème s'est accentué lorsque face au quasi-monopole de la suite Office, qui utilisait des formats de fichiers non documentés, fermés, et limitant tout possibilité d'interopérabilité, l'ISO[16] chercha à pousser un format de fichier standardisé et ouvert. Le risque pour Microsoft étant de devoir implémenter pour sa suite bureautique des formats qu'il ne contrôlerait pas et que ses concurrents, notamment les solutions de bureautique Open Source, maîtriseraient autant que lui. Microsoft dû donc se résoudre à proposer un format ouvert.

[14] An Open Letter to Hobbyists, homebrew Computer Club Newsletter Volume 2, Issue 1

[15] Interview du Chicago Sun-Times du 1e juin 2001

[16] Organisation internationale de normalisation, chargé de définir des normes internationales pour les domaines industriels et commerciaux (norme ISO 9001 pour le management de la qualité, 27001 pour le management de la sécurité des systèmes d'information, ...)

La transition vers l'Open Source

La vraie transition vers *l'Open Sour*ce de Microsoft s'entame non pas à cause de la qualité supérieure des méthodes *Open Source*, mais face à l'émergence du *Cloud*. En effet, Linux et les logiciels *Open Source* sont dorénavant la norme pour le déploiement de serveurs et de services en ligne. La concurrence actuelle n'est plus uniquement sur la fourniture de logiciels mais également sur la fourniture de services d'infrastructure, le *Cloud,* et le concurrent de Microsoft sur ce terrain de jeu s'avère être Amazon, avec une très sérieuse avance.
Si les premières versions des services d'Azure, la plateforme Cloud de Microsoft, étaient tournées vers la fourniture de solutions de développement liées à l'écosystème de Microsoft, l'ouverture d'un service à même de concurrencer Amazon impliquait de faire un pas supplémentaire vers Linux et les communautés Open Source.
Après avoir déclaré en 2014 par la voix de son nouveau CEO, Satya Nadella - qui avant d'être nommé à la tête de la société était vice-président de la branche *Cloud* et Entreprise de la société - que "Microsoft aime Linux", Microsoft est devenu sponsor premium - le plus haut niveau - de *The Linux Foundation*, la fondation encadrant non seulement le projet Linux, mais également les projets open source majeurs permettant de gérer les infrastructures Cloud.

Microsoft n'est d'ailleurs pas le seul acteur hégémonique à soutenir la fondation Linux : parmi les sponsors premium se trouvent Google, Huawei et Tencent, ainsi que Facebook, Alibaba, Baidu et WeBank ; et se retrouvent au conseil d'administration des représentants de Google, Facebook, Huawei et Microsoft, entre autres.

Pour asseoir sa position d'acteur impliqué dans les communautés *Open Source*, Microsoft a également annoncé le rachat de GitHub en 2018, la plus importante plateforme d'hébergement de codes sources publics et privés.

Conclusion

Le rôle du logiciel a été fortement modifié entre l'avènement de l'ordinateur personnel et l'ère actuelle du *Cloud*. L'émergence de différents modèles, issus de cultures différentes, a amené les acteurs à évoluer drastiquement.

Alors que l'environnement technologique laissait à penser que seule importait la capacité à apporter des solutions techniques viables, les questions philosophiques et morales se sont invitées dans le débat. Pour autant, le succès de l'*Open Source* est lié justement à sa capacité à s'éloigner du débat idéologique et moral, pour se focaliser sur des aspects pratiques de qualité de la solution, mais également pour tirer bénéfice d'une forte capacité de rassemblement.

En effet, ce qui anime ces communautés *Open Source*, c'est l'envie de résoudre des problèmes techniques, un mélange de dépassement de soi et de recherche de gloire, où la transparence liée au partage du code source permet de prouver sa compétence face à sa communauté. Cette capacité d'émulation est essentielle pour les acteurs qui y évoluent car au-delà des solutions techniques qui sont ainsi créées, c'est également un vivier de talents extrêmement important. Enfin, une solution technique, si elle est poussée correctement au sein de la communauté, peut influencer le marché en devenant un standard de fait.

L'intégration au sein des communautés *Open Source* est donc essentiel pour tenter d'y exercer une influence. Or, si on regarde le paysage français, on ne peut que constater un manque d'intégration au sein des communautés *Open Source*.
Les grands éditeurs français ont gardé la position de Microsoft des années 90 et ne s'impliquent dans aucune communauté. En face la majorité des regroupements sont des associations prônant le libre plutôt que *l'Open Source*, intégrant la dimension morale : le Conseil National du Logiciel Libre, AFUL[17], ADULLACT[18], Framasoft et April sont des défenseurs de la philosophie de la *Free Software Foundation*, ce qui ne favorise pas le rapprochement avec des éditeurs, qui de leur côté craignent de perdre la main sur leur propriété intellectuelle.
Il est donc essentiel en France de faire preuve de pragmatisme et d'intégrer que *l'Open Source* et le libre sont des concepts proches mais différents, et qu'un peu de pragmatisme économique est essentiel.

[17] Association Francophone des Utilisateurs de Logiciels Libres
[18] Association des Développeurs et Utilisateurs de Logiciels Libres pour les Administrations et les Collectivités Territoriales

Webographie

- Copyright Timeline: A History of Copyright in the United States - Association of research libraries
- Copyright of Computer Programs - digital-law-online.info
- Final Report of the National Commission on New Technology Uses of Copyrighted Works - Chapter 3, Computers and Copyright
- Legifrance
  - Journal Officiel du 4 juillet 1985, loi n° 85-660 du 3 juillet 1985 relative aux droits d'auteur [...]
    - Titre Ier - "DU DROIT D'AUTEUR"
    - Titre V - "DES LOGICIELS"
  - Journal Officiel du 17 janvier 1982, enrichissement du vocabulaire de l'informatique
  - Article L113-9 du Code de la propriété intellectuelle modifié par la Loi n° 94-361 du 10 mai 1994 - art. 2
- Copyright : définition juridique et différence avec le droit d'auteur - jurifiable.com
- Licences libres - aful.org
- Qu'est-ce que le logiciel libre - gnu.org
- Qu'est-ce que le copyleft - gnu.org
- Petit guide sur les logiciels libres à l'intention des professionnels et des juristes - Organisation Mondiale de la Propriété Intellectuelle (via archive.org)
- Libre comme Liberté, Chapitre XI "Open Source" - wikisource.org
- A brief history of Lean - lean.org
- Business Software Alliance - Membres
- Liste des licences libres, open sources et collaborative - spdx.org
- "An Open Letter to Hobbyists" - digitalbarn.com
- Is Microsoft the Great Satan? - gnu.org (via archive.org)
- Why Boycott Microsoft - vcnet.com (via archive.org)
- Ballmer: 'Linux is a cancer' - linux.com, 01/06/2001
- Microsoft aime Linux - zdnet.fr, 30/10/2004
- The Linux Foundation - Corporate Membres
- Open Source Initiative - Sponsors & Support
- April - Adhérents personnes morales
- CNLL - Membres
- L'avenir incertain des licences open source - Le Mag IT, 06/03/2020