# Partie 2 : Systèmes Ouverts et Environnements de Développement

## 2.1 Historique, motivation et objectifs

Un logiciel propriétaire est un logiciel dont le contenu appartient non pas à ceux qui l'ont mis au point mais à la société qui les emploie. L'utilisateur d'un logiciel propriétaire peut seulement l'utiliser ; il ne peut pas en étudier le fonctionnement, le modifier ou le copier.

À l'inverse du logiciel propriétaire, un logiciel libre permet à ses utilisateurs d'étudier en détail son code-source, de le modifier, de le copier et de rendre publiques les modifications qu'ils ont opérées dessus.

### 2.1.1 Idée de base

L''idée de Logiciel Libre a été popularisé par Richard Stallman depuis 1984, année où il a créé la Free Software Foundation. Conscient qu'il est impossible d'utiliser un ordinateur sans système d'exploitation et que sans système d'exploitation libre, il est obligatoire d'utiliser des logiciels propriétaires, il démarre le premier projet de la fondation, le projet GNU « GNU's Not UNIX » (littéralement, « GNU n'est pas UNIX »). Ce projet vise à concevoir un système d'exploitation complet et entièrement libre. Ce système sera compatible avec UNIX, mais sera différent. Aujourd'hui ce système existe, et s'appelle GNU/Linux.

Pour valider ce système, une base légale est nécessaire. Cette base légale est la GNU GPL, pour GNU General Public Licence. La GNU GPL est la licence des logiciels libres par excellence. Elle détermine des conditions de distribution qui garantissent les libertés de l'utilisateur. Un programme protégé par la GPL est libre, mais la GPL impose aussi que tout travail dérivé de ce logiciel reste libre.

Il a intitulé cette licence « gauche d'auteur » (copyleft), car au lieu d'interdire, elle donne le droit de copier.

Sigle de Copyleft

Stallman a lui-même développé des travaux en matière de logiciels libres, tels que le compilateur de langage C du projet GNU, et l'éditeur de texte Emacs. Ses travaux ont inspiré de nombreux autres développeurs à proposer du logiciel libre selon les conditions de la GPL.

Afin de faciliter le travail collectif entre les différents développeurs bénévoles, Stallman créa en 1985 la fondation du logiciel libre (Free Software Foundation). Beaucoup d'outils pour le système d'exploitation GNU ont été développés au sein de cette fondation, en particulier la librairie C et l'interface shell.

Logo du projet GNU

En 1990, tous les composants nécessaires pour GNU ont été développés à l'exception du noyau. Un premier noyau appelé HURD a été élaboré mais il a rencontré beaucoup de problèmes. En 1991 ; Linus Torvalds, un étudiant informaticien de l'université Helsinki en Finlande a conçu sous la licence GPL un noyau appelé LINUX. Linus lui-même, ne savait pas que le travail qu'il a effectué durant son temps libre allait changer le visage de l'industrie informatique dans le monde entier.

Les premières versions de Linux ont attiré beaucoup de développeurs à travers le monde ce qui a fait évolué ce système avec une très grande vitesse, ce qui a rendu Linux le plus grand projet collectif dans l'histoire de l'humanité.

En constant le grand succès de Linux, Stellman a décidé de laisser tomber le noyau HURD et d'adopter à sa place le noyau Linux pour GNU ; ce qui donne naissance à GNU/LINUX.

### 2.1.2 Définition du logiciel libre

Richard Stallman défini le logiciel libre en trois mots : ***Liberté, égalité, fraternité***

***Liberté*** : car vous faites ce que vous voulez avec le programme, pas ce que le développeur a décidé pour vous
***Égalité*** : car chacun possède les mêmes libertés face au logiciel, le développeur n'est pas tout puissant
***Fraternité*** : car le logiciel libre encourage la collaboration entre les utilisateurs, la possibilité d'échanger, de partager

### 2.1.3 Les 4 libertés fondamentales

Free Software Foundation (FSF) donne la définition du logiciel libre basée sur quatre libertés:
***Liberté 0 :*** Liberté d'exécuter le programme, pour tous les usages
***Liberté 1 :*** Liberté d'étudier le fonctionnement du programme (comprend l'accès au code source)
***Liberté 2 :*** Liberté de redistribuer des copies (comprend la liberté de vendre des copies)
***Liberté 3 :*** Liberté d'améliorer le programme et de publier ses améliorations

- ✓ Suppose l'accès au code source
- ✓ Encourage la création d'une communauté de développeurs améliorant le logiciel
- ✓ Permet le *fork*, c.à.d. la possibilité de création d'une branche de développement concurrente (notamment en cas de désaccord entre développeurs)

Ces 4 libertés sont faites pour protéger l'utilisateur final et ne sont pas des obligations

Un logiciel libre est un logiciel qui respecte les 4 libertés fondamentales du logiciel libre
En informatique, beaucoup de logiciels sont distribués sans leur code source, et il est interdit d'essayer de comprendre leur fonctionnement (on parle dans ce cas d'un logiciel privateur). Il est interdit de les partager avec vos amis, et il est interdit d'essayer de les modifier pour les adapter à vos besoins.

En revanche, un logiciel libre vous garantit les quatre libertés. Avec un logiciel libre, c'est comme ci on a le plat, la recette, le droit de redistribuer (ou de vendre) le plat, la recette, et même de la modifier.

Un logiciel libre n'est pas forcément gratuit. Ambiguïté en anglais de (free = gratuit = libre). Aussi, un logiciel gratuit n'est pas forcément libre.
Donc, un logiciel libre peut être payant, un logiciel gratuit est rarement un logiciel libre.

Un logiciel libre ne doit rendre son code source public que pour ces utilisateurs.

Le "Logiciel non libre" est le contraire d'un "Logiciel Libre". On dit qu'il est propriétaire, privateur ou privatif.

### 2.1.3 Communauté du Logiciel Libre et le publique visé

La qualité du logiciel est souvent proportionnelle au nombre de développeurs. Plus la communauté de développement s'étend, plus elle devient une preuve de qualité et de réactivité. De la même manière, la communauté des utilisateurs, ayant comme rôle principal de faire remonter des dysfonctionnements et des suggestions, a une influence proportionnelle à sa taille

La communauté du Logiciel Libre est constituée de programmeurs, utilisateurs, traducteurs et graphistes. Elle est importante et réactif.

La ***relation*** entre utilisateur et développeur n'est plus une relation de client à fournisseur, mais de ***personne à personne***, privilégiant l'entraide

Le libre s'adresse à une grande variété de clients :

- ✓ les particuliers : Le particulier trouvera dans le logiciel libre une alternative crédible aux solutions propriétaires. Il y gagnera :

- en coûts
- en sécurité (ex : Linux est peu sensible au virus et permet la gestion de l'accès au fichiers) et en protection de la vie privée (le code source étant disponible, il est impossible de lui adjoindre des spyware à l'insu du client).

✓ les organismes publics : Les organismes publics y gagnent :
- en coûts
- en pérennité (le fonctionnement informatique de l'administration doit être indépendant des intérêts commerciaux : l'ouverture du code source garantit ***la continuité*** du maintien d'un logiciel libre)
- en sécurité (les données sont sécurisées ; l'ouverture du code garantit qu'aucune faille n'existe, permettant d'accéder frauduleusement à l'information.

✓ les entreprises

✓ les pays moins riches : Les logiciels libres sont accessibles pour les pays en développement, car le plus souvent, ils sont gratuits ou de prix extrêmement modique.

### 2.1.4 Fonctionnement du logiciel libre

Depuis l'origine de l'informatique, il y a toujours eu des utilisateurs qui considèrent l'informatique comme une passion. De nombreuses personnes peuvent passer des journées (et même des nuits) sur un ordinateur juste pour le plaisir de développer des programmes.

Avec l'arrivée d'Internet, ces passionnés d'informatique ont pu partager leurs connaissances et développer en commun des programmes de plus en plus performants. De plus, depuis quelques années de nombreuses sociétés dont la plus connue est IBM, sont venues rejoindre les rangs des bénévoles ce qui permet aux logiciels libres de se développer encore plus vite.

Le libre peut être considéré comme un retour naturel au partage du savoir qui se situe dans la tradition du travail scientifique. Aux débuts de l'informatique, les logiciels étaient diffusés avec leur code source, ce qui a permis aux chercheurs de faire évoluer les programmes en fonction de leurs besoins.

### 2.1.5 Contrôle par l'utilisateur

Un des objectifs principaux du logiciel libre est de permettre à l'utilisateur d'avoir le contrôle sur son ordinateur et sur les logiciels qu'il utilise. Ce contrôle est donné individuellement : chacun peut étudier en détail ce que fait le logiciel, et le modifier s'il le souhaite. Mais les utilisateurs ont aussi le contrôle de manière collective sur leur ordinateur : on ne peut étudier l'ensemble des logiciels que l'on utilise, mais on peut être certain que si un logiciel contient une fonctionnalité cachée ou malveillante, elle sera découverte un jour et un correctif sera proposé. Ceci a de plus, pour effet de dissuader, la plupart du temps, les développeurs d'ajouter de telles fonctionnalités.
Les propriétaires de logiciels propriétaires (les développeurs, ou l'entreprise pour laquelle ils travaillent) ont le pouvoir d'espionner ou de restreindre les utilisateurs.

La définition du logiciel libre par la FSF précise :
« Quand les utilisateurs ne contrôlent pas le programme, c'est le programme qui contrôle les utilisateurs. Le développeur contrôle le programme, et par ce biais, contrôle les utilisateurs. Ce programme non libre, ou «privateur», devient donc l'instrument d'un pouvoir injuste. »

### 2.1.6 Les alternatives libres

Les logiciels libres sont disponibles sur pratiquement tous les systèmes d'exploitation (libres ou pas), par exemple :

***Suite bureautique :***
- ✓ Propriétaire : Microsoft Office
- ✓ Libre : LibreOffice, OpenOffice.org

***Navigateur :***
- ✓ Propriétaire : internet Explorer, Google chrome
- ✓ Libre : Mozilla, Firefox, SeaMonkey

***Multimédia :***

***Lecteur multimédia***

- ✓ Propriétaire : Windows Media player
- ✓ Libre : MPlayer, Xine, Totem, Kaffeine, VLC

***Dessin assisté par ordinateur***

- ✓ Propriétaire : Adobe Photoshop, Paint Shop Pro, Open Canvas, Corel Photo Paint
- ✓ Libre : Gimp, Krita, Tux Paint, InkScape, Sodipodi

***Les systèmes d'exploitation***

- ✓ Propriétaire : Microsoft windows, Mac OS...
- ✓ Libre : GNU/Linux (Debian, Gentoo, Ubuntu, Fedora, Mandriva...) , BSD (FreeBSD...)

***Serveurs Web:***

- ✓ Propriétaire : Internet Information Services (IIS ), Sun Java System Web Server
- ✓ Libre : Apache, Tomcat

***SGBD:***

- ✓ Propriétaire : Oracle Database, Microsoft SQL Server
- ✓ Libre : MySQL, PostgreSQL

***Programmation:***

- ✓ Propriétaire : Pascal, Delphi
- ✓ Libre : Perl, Python, PHP

***Antivirus:***

- ✓ Propriétaire : BitDefender, Kaspersky Anti-Virus, Norton AntiVirus
- ✓ Libre : ClamWin AntiVirus, Winpooch (ClamWin ne dispose pas de protection en temps réel. Toutefois cette fonctionnalité peut être assurée par un logiciel tiers qui travaillera de concert avec ClamWin : Clam Sentine)

---

Il existe donc des milliers de logiciels libres toutes catégories confondues. Nul doute que nous pouvons trouver aisément une solution libre à notre goût.

## 2.2 Le logiciel « open source »

La définition du logiciel libre pose un problème due à la nature même de la langue anglaise et qui a été utilisée dans l'écriture de la licence GPL. En effet le terme « free » peut avoir deux sens : « gratuit » et «libre ».
En 1998, la compagnie Netscape a voulu libéré le code source de son navigateur Mozilla. Pour cela, elle a fait appel à Eric Raymond est qui est l'un des pionniers du mouvement logiciels libres. Ceci a donné naissance à l'initiative OSI (Open Source Initiative).

Au lieu du terme « Free », OSI préfère utiliser le terme « Open » qui dénote la disponibilité et la modificabilité du code source.


La désignation open source, ou « code source ouvert », s'applique aux logiciels dont le code source est disponible et la licence respecte des critères précisément établis par l'Open Source Initiative :

- ✓ Libre redistribution
- ✓ Accès au code source
- ✓ Travaux dérivés

Sil n'y a pas d'accès au code, cela implique pas de vérification possible sur ce que fait le logiciel. Il faut faire une confiance aveugle au développeur alors que l'expérience montre que les abus sont nombreux (failles de sécurité, espionnage).

Souvent, un logiciel libre est qualifié d'Open Source, car les licences compatibles open source englobent les licences libres selon la définition de la FSF.

Le terme open source est en concurrence avec le terme « free software » recommandé par la FSF.

### 2.2.1 Free Vs Open

- **Free software** → Free Software Foundation (FSF)

FSF fait référence directe à la liberté
**www.fsf.org**

- « logiciel libre ne signifie pas gratuit ».

- **Open source software** → Open Source Initiative (OSI)

(**www.opensource.org**)

- Un logiciel peut être Open source mais pas gratuit (cas du projet Darwin d'Apple, un système d'exploitation de base pour attirer la communauté attachée à Linux)

La meilleure façon de savoir dans quel domaine se place le logiciel est de se référer à la licence d'utilisation sous laquelle est distribué le logiciel.

### 2.2.2 Ouvert Vs gratuiciel Vs partagiciel

Il ne faut pas confondre les logiciels ouverts avec :
***Les gratuiciels (freewares):*** Un logiciel freeware est un logiciel propriétaire gratuit. Le terme freeware désigne des logiciels gratuits qui peuvent être ni ouverts, ni libres. Ce logiciel est mis gratuitement à disposition par son créateur soit en tant que logiciel libre, soit en tant que logiciel propriétaire, auquel cas il est soumis à certaines contraintes quant à sa diffusion.

***Les partagiciels (sharewares) :*** logiciel propriétaire, protégé par le droit d'auteur, qui peut être utilisé gratuitement durant une certaine période ou un certain nombre d'utilisations. Durant la période d'utilisation gratuite, il est possible que certaines fonctions du logiciel ne soient pas disponibles

***Les logiciels tombés dans le domaine public:*** Selon les pays, après une certaine période les droits de propriété intellectuelle tombent et la création intellectuelle devient une propriété publique

### 2.2.3 Les bénéfices de l'open source pour le client

Bien sûr, les bénéfices économiques sont parmi les premières raisons dans le choix de solutions open source. Même si « libre ne signifie pas gratuit », ces solutions ont toujours un coût de possession sensiblement moins élevé que leurs équivalents propriétaires.

Les principaux arguments du choix de l'open source sont :

- La non-dépendance, ou moindre dépendance, par rapport à un éditeur particulier.
- L'ouverture est également un argument de poids. Les solutions open source sont en général plus respectueuses des standards, et plus ouvertes vers l'ajout de modules d'extension.
- La pérennité est un autre critère de choix fort. Une solution open source leader offre une garantie de pérennité (continuité) supérieure à la majorité des solutions propriétaires.
- Et la qualité, car dans beaucoup de domaines les solutions open source sont réellement, et objectivement, supérieures. Le très grand nombre de déploiements et donc de retours d'expérience,

mais aussi leur modèle de développement et leur intégration de composants de haut niveau, permet à beaucoup de surclasser les produits propriétaires.

L'accès au code source permet donc :

- ✓ de ***comprendre*** parfaitement le fonctionnement du programme
- ✓ ***d'améliorer*** le programme
  - o Corriger les bugs
  - o Ajouter des fonctionnalités
- ✓ de "***porter***" le programme sur d'autres plateformes

## 2.3 Licence et Aspects légaux

A 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.

Le concepteur de la GPL (licence publique générale), Eben Moglen, insiste sur la distinction entre licence et contrat : une ***licence*** est une autorisation unilatérale, tandis qu'un ***contrat*** suppose des obligations réciproques. Les logiciels libres sont distribués avec de simples licences. Généralement, ils sont également distribués sans la moindre garantie.

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

***Domaine public***

Il ne s'agit pas réellement d'une licence, mais simplement du fait que le logiciel n'a aucun ayant-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
Toutefois, la durée de protection des droits d'auteur est bien plus longue que le plus ancien des logiciels
On ne trouve donc dans le domaine public que des logiciels qui y ont été mis.

***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), X Window (licence MIT) et Apache Software Foundation (licence Apache).

***Copyleft, 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. L'inverse est impossible
Le degré de liberté moindre des licences de type copyleft est critiqué par des acteurs des projets BSD et des acteurs commerciaux.

**Liste des Licences GNU :**

Il existe plusieurs licences GNU, telles que : GPL, Microsoft Public License, Intel Open Source License, FreeBSD license, BitTorrent Open Source License, Apache License, etc.
(http://www.gnu.org/licenses)

**Autres types de licences**
***Freewares, logiciels gratuits***
On utilise le terme freeware pour les logiciels propriétaires qui sont distribués gratuitement
Les freewares ne sont pas libres car leur code source n'est pas disponible et donc seul l'auteur original peut l'améliorer et publier des versions modifiées
En outre, la revente d'un freeware est souvent restreinte. Les freewares sont de plus en plus rares et sont souvent considérés comme une sorte de frein au logiciel libre
À l'inverse, les logiciels libres ne sont pas forcément gratuits. Dans les faits, la plupart des logiciels libres se trouvent gratuitement sur Internet, mais peuvent simultanément être achetés à un prix comparable aux logiciels propriétaires. Dans ce second cas, l'achat donne généralement droit à une garantie de support.

***Sharewares***
Le shareware est un logiciel qu'on peut légitimement se procurer gratuitement, mais qu'on doit payer si l'on désire l'utiliser
Aucun logiciel libre n'est un shareware
A la limite, un logiciel libre pourrait encourager la rétribution de l'auteur, sans que cela n'ait force de licence (sans quoi il ne serait plus libre). Mais toute personne serait libre de supprimer l'encouragement et de redistribuer cette version allégée.

***Shared source***
Le terme Shared source (code source partagé) vient de la Shared Source Initiative de Microsoft. Il s'agit d'un type de licence qui donne le droit de regarder le code source et parfois d'en distribuer des versions modifiées
Toutefois, le droit de vendre n'est pas donné et les licences shared source ne sont pas considérées comme des licences de logiciel libre

## 2.4 Avantages et inconvénients des systèmes logiciels ouverts

### 2.4.1 Avantages

Les systèmes logiciels ouverts présentent plusieurs avantages, on cite :

- ✓ ***Plus sûrs*** : Code source disponible ; pas de backdoors ou espions cachés (Clé de cryptage NSAKey (National security Agency) découverte dans le code du windows NT4 par un canadien)
- ✓ ***Plus stables*** : Plus de programmeurs motivés, une course à la découverte des failles (logiciel propriétaire: le profit prime et la découverte des failles n'est pas une motivation des programmeurs mais de leurs responsables)
- ✓ ***Moins chers*** : Même en cas de paiement, ils restent moins chers que les logiciels propriétaires. Pratiquement pas de frais de maintenance
- ✓ ***Meilleure conformité aux standards*** : Les logiciels propriétaires retiennent les utilisateurs à leurs produits. La conformité aux standards signifie qu'il est possible de choisir un autre logiciel pour d'autres raisons que le fait qu'il soit compatible avec ce dont en dispose
- ✓ ***Utilisation et modification immédiate*** (dès l'acquisition) : On n'attend pas la réaction d'un grand fournisseur pour un changement, on peut l'opérer immédiatement ou demander au autres de le faire (communauté du libre)
- ✓ ***Très peu de restrictions quant à leur utilisation*** : Les licences donnent beaucoup de liberté contrairement aux contrats des logiciels propriétaires qui parfois changent les termes du contrat après son acceptation et obligent les utilisateurs à accepter ces changements à l'avance (Microsoft)

### 2.4.2 Inconvénients

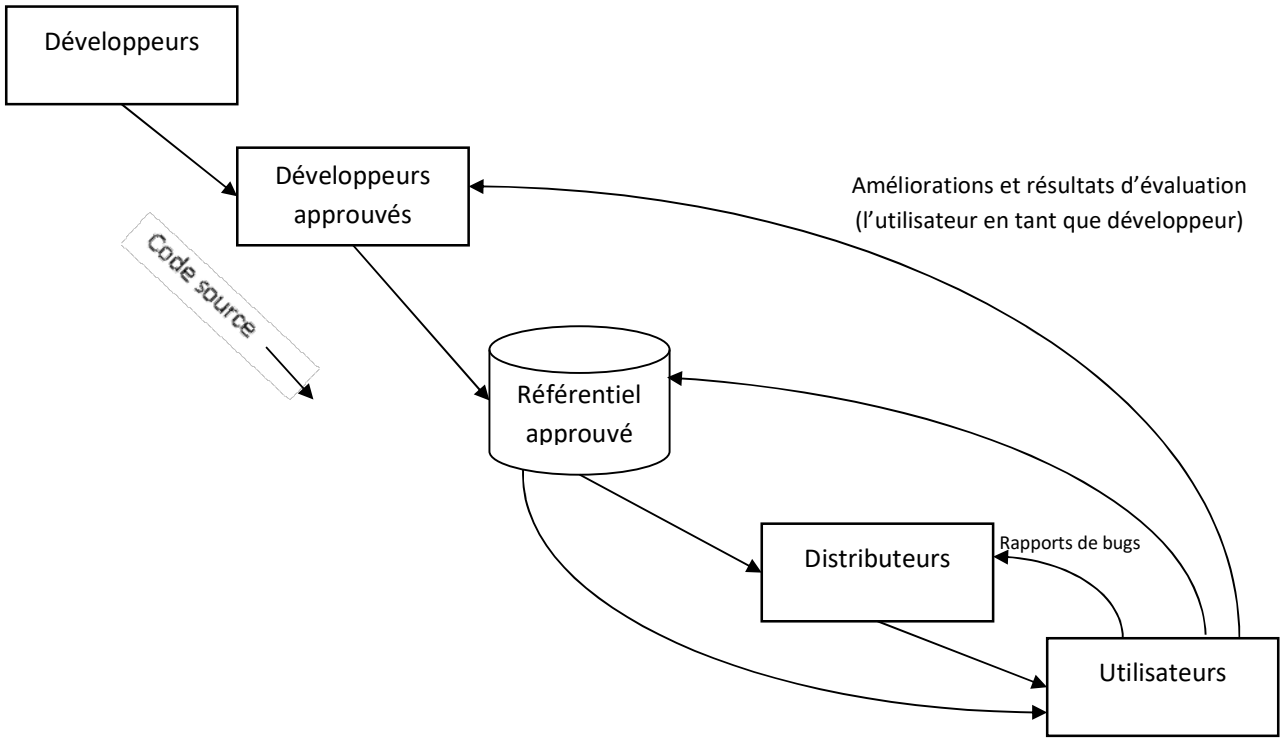
Parmi les inconvénients généralement évoqués par ceux qui sont contre les systèmes ouverts, on cite :

- ✓ ***Déséquilibre le statut quo du marché et menace la propriété intellectuelle*** (argument utilisé mais pas correct)
- ✓ ***Les systèmes sont ouverts aux manipulations*** : Les hackers et les criminels peuvent y introduire ce qu'ils veulent (Pas vrai en totalité : un système analysé et examiné par des centaines de personnes est moins susceptible de comporter des backdoors)
- ✓ ***Manque de support*** : vers qui se tourner en cas de besoins si un logiciel est développé par des centaines de programmeurs à travers le monde?

## 2.5 Développement des systèmes ouverts

Il n'existe aucun modèle de cycle de vie normalisé pour le développement de logiciels open-source. Néanmoins, différents chercheurs et développeurs ont proposé différents modèles de cycle de la vie pour le développement en rapport avec leurs propres expériences de développement, leurs besoins et leurs applications.

Parmi ces modèles ; on trouve celui proposé par le département de défense des Etats-Unis.




Ce cycle de vie comporte :

***Les développeurs***: Ils comprennent toutes les personnes qui ont initié et qui ont extrêmement contribué dans le développement initial du logiciel.

***Les développeurs approuvés*** : Ils comprennent les personnes qui contribuent en continu dans le développement de ce logiciel et qui ont gagné la confiance des initiateurs à travers leur implication en

continu dans le processus de développement. Ils sont donc devenus une partie de la communauté des développeurs du noyau. Ces développeurs sont autorisés à faire des mises à jour et des changements directement dans le référentiel approuvé. L'ensemble d'utilisateur et de distributeurs envoient leurs demandes de changements et de mises à jour aux développeurs approuvés.

***Le référentiel approuvé***: Dans le développement du logiciel open-source, le référentiel spécifie la maison d'où toutes les informations relatives à ce logiciel peuvent être récupérées. L'utilisateur et les développeurs approuvés peuvent accéder directement au référentiel ou à travers le distributeur. Le référentiel approuvé spécifie donc l'espace d'où on peut obtenir la version officielle du logiciel et d'autres renseignements connexes tels que le rapport de bugs, le journal des modifications, la documentation, etc.

***Les distributeurs***: C'est les personnes qui possèdent la copie du logiciel développé et qu'ils peuvent l'utiliser pour effectuer d'autres tâches telles que la modification, l'intégration, les tests, la configuration, etc.

***Les utilisateurs***: Ce sont ceux qui utilisent le logiciel. L'utilisateur peut être classé en tant qu'utilisateur passif ou actif. Les utilisateurs passifs sont ceux qui téléchargent le logiciel pour une utilisation ou une étude et qui n'ont jamais participé au développement. L'utilisateur actif participe au processus de développement en effectuant des tâches, telles que trouver des bugs, donner des avis sur le logiciel, etc.

Dans ce processus de développement le flux du code source suit une approche top-down du développeur - développeur approuvé - référentiel approuvé - distributeur – utilisateur, alors que les évaluations et les rapports de bugs suivent une approche bottom-up.

## 2.6 Tendances futures

Au cours des dix dernières années, les attitudes envers l'open source dans les environnements informatiques d'entreprise ont radicalement changé, les utilisateurs étant passés du scepticisme quant à sa capacité à faire face aux charges de travail critiques à son déploiement à grande échelle sur leurs infrastructures. Il domine désormais la couche du système d'exploitation dans les infrastructures informatiques et devient rapidement partie intégrante de la plupart des initiatives informatiques stratégiques.

A titre indicative sur l'utilisation du logiciel libre, voici quelques résultats d'après l'étude GARTNER :

- ✓ En 2014, 30% des applications fonctionnant sous UNIX propriétaire ont migré vers LINUX
- ✓ En 2016, 50% des principales organisations non IT ont utilisé les logiciels libres pour gagner un avantage compétitif.
- ✓ En 2021, le rapport de recherche souligne également que 89 % des organisations ont déclaré utiliser un logiciel de SGBD gratuit ou open source ; un tiers a déclaré l'utiliser dans plus de 50 % de leur organisation. »
- ✓ Gartner prédit que d'ici 2022, plus de 70 % des nouvelles applications internes seront développées sur une base de données open source, et 50 % des instances de bases de données relationnelles propriétaires existantes auront été converties ou seront en cours de conversion.

C***obal 2000*** : C'est un classement annuel des 2000 plus grandes entreprises mondiales publié par le magazine américain Forbes. C'est un indicateur pertinent des entreprises qui dominent le marché mondial, mais reste une interprétation parmi beaucoup d'autres.

## 2.7 Environnements de développement ouverts

Un environnement de développement est un ensemble d'outils pour augmenter la productivité des programmeurs qui développent des logiciels. Un environnement de développement comporte typiquement les outils nécessaires pour analyser, écrire, et déboguer un programme.

Un environnement de développement peut également comporter les outils suivants :

- ✓ un outil de création d'interface graphique. Tel outil permet au programmeur de gagner un temps significatif dans la construction de l'interface graphique de son programme. Jusqu'à l'arrivée de la technologie Java, de tels outils ciblaient toujours un système d'exploitation en particulier.
- ✓ un outil pour réaliser automatiquement des tests.
- ✓ des outils d'analyse du code source. Par exemple un générateur de graphique qui permet d'obtenir le diagramme en arbre de l'utilisation d'une fonction du programme.
- ✓ un moteur de recherche qui tient compte du langage de programmation: il permet par exemple de rechercher le nom d'une fonction en évitant les commentaires et les expressions littérales, et en se limitant au cadre d'un module ou d'une classe.
- ✓ des outils destinés à assister aux opérations préliminaires à la programmation, par exemple des outils de modélisation, ou d'analyse des exigences.
- ✓ un outil de contrôle de versions. Un tel outil permet à plusieurs programmeurs de travailler simultanément sur les fichiers de code source du programme.
- ✓ Les environnements de développement intégré sont parfois issus d'outils qui offrent une seule et unique fonction et que leurs auteurs ont ensuite enrichis et fusionnés avec d'autres outils. Il peut s'agir d'outil de création d'interface graphique ou de manipulation de base de données.

Un environnement de développement intégré est un ensemble d'outils destinés à programmer dans un langage donné, qui sont distribués ensemble. Il permet de manipuler les outils de programmation depuis une interface graphique simplifiée.

Voici une liste de quelques environnements de développement ouverts :

- ✓ ***Anjuta*** : environnement de développement intégré (IDE) pour le C et le C++ sur GNU/Linux.
- ✓ ***Code::Blocks*** : environnement de développement multiplate-forme pour C, C++ et D.
- ✓ ***Dev-C++*** : environnement de développement C et C++ pour Windows utilisant le compilateur GCC.
- ✓ ***Eclipse*** : environnement de développement multiplate-forme
- ✓ ***KDevelop*** : suite de développement intégrée à KDE.
- ✓ ***NetBeans***
- ✓ Ect.