



# Gardien Virtuel

s é c u r i t é  
i n f o r m a t i q u e

## Guide sur les Logiciels

**Par:**

Patrick Boucher, CISSP, CISA

Droit d'auteur 2005 par Gardien Virtuel, Ce document est ouvert au public. La distribution de version modifié de ce document est interdite sans l'autorisation écrite de Gardien Virtuel.

La dernière version de ce document se trouve à l'URL suivant:

<http://www.gardienvirtuel.com/wp/guidell.pdf>

### Détails du document

Type:	Guide et positionnement
Date de création:	Le lundi 20 juin 2005
Dernière modification:	Le mardi 18 avril, 2006
Version:	1.1 : Version finale

## Table des matières

Introduction.....	3
Logiciel Tablette.....	3
Logiciel commercial ou propriétaire.....	3
Logiciel au code source libre.....	3
Une définition des logiciels libres.....	4
Code Source.....	4
Logiciel libre et logiciel commercial.....	5
Modèle de Logiciel Libre.....	6
Utilisation des logiciels libres dans l'entreprise.....	7
Survol de l'industrie du Logiciel Libre.....	8
Préoccupation par rapport au logiciel libre.....	8
Coûts des licences.....	8
Disponibilité de support.....	8
Support interne.....	9
Support externe.....	9
Maximiser la flexibilité des options de supports.....	9
Fiabilité du logiciel.....	10
Maturité et longévité.....	10
Scénario de développement.....	11
Développement maison.....	12
Développement externe ou en impartition.....	13
Développement accidentel.....	13
Continuité de développement.....	13
Analyse et gestion du risque.....	14
Disponibilité des compétiteurs viables.....	14
Garanties et indemnités.....	14
Cycle de vie du logiciel.....	15
Première phase : Développement de la solution.....	15
Deuxième phase : Première adoption.....	15
Phase long terme ou de maintenance.....	16
Licences d'utilisation.....	17
GNU General Public Licence (GPL).....	17
Berkeley Software Distribution (BSD).....	17
Référence en logiciel libre.....	18
Bibliographie.....	18

## **Introduction**

L'objectif de ce guide est de fournir l'information nécessaire aux gestionnaires et chefs de projets, afin de mieux comprendre, analyser, planifier et obtenir une vue d'ensemble des processus de développement ou d'acquisition de logiciel libre. Ce document ne se veut pas une directive ou une argumentation en faveur du logiciel libre. Ce document contient l'information disponible et courante dans le marché des solutions logiciels.

### ***Logiciel Tablette***

Un logiciel tablette est une application propriétaire et grand publique. Cette application se veut la plus générale possible afin de répondre aux besoins du plus grand nombre d'organisation. Habituellement pour le prix d'acquisition de l'application, l'acheteur obtient une copie physique ainsi qu'un droit limité d'utilisation de l'application.

À cela s'ajoute les frais de support et les frais de licence (si plus d'une personne utilisent la même application).

Exemple: Simple comptable ou fortune 1000 s'achètent dans plusieurs magasins informatiques.

### ***Logiciel commercial ou propriétaire***

Il est dit d'un logiciel qu'il est « commercial » ou « propriétaire » si celui-ci est la propriété exclusive d'une personne, qui charge des frais de licence pour l'utilisation, la distribution et la modification de la solution.

De plus le terme propriétaire est utilisé dans les cas où les utilisateurs n'ont pas accès aux codes source.

Le format de document Microsoft Word « .DOC » est propriétaire puisque puisque Microsoft ne divulgue pas la recette ou le fonctionnement interne des .DOC

### ***Logiciel au code source libre***

Les logiciels libres ont des attributs particuliers, ce qui les différencient des logiciels produit traditionnellement par les entreprises. Il est dit d'un logiciel qu'il est au code source libre lorsque l'accès à son code source est permis et voulu.

Le terme libre vient du fait que l'utilisateur a la liberté de faire ce qu'il désire de l'application. Celle-ci n'est pas toujours gratuite. Mais l'utilisateur a toujours la possibilité de la modifier et de l'utiliser librement. La valeur principale des logiciels libres par rapport aux commerciaux est l'ouverture des formats et des techniques utilisées par les logiciels. Ces « formules » sont ouvertes à l'analyse et à la modification par ses utilisateurs.

## Une définition des logiciels libres

Les logiciels libres sont des applications informatiques ayant des particularités relatives aux droits d'exploitations et aux cadres légaux qui les entourent.

Présentement, le groupe « Open Source Initiative » s'occupe de définir ce qu'est un logiciel libre.

<http://www.opensource.org/>

Les critères sont comme suit :

- Possibilité d'utiliser le logiciel pour n'importe quel usage
- Effectuer des copies sans réserve
- Avoir accès et modifier le code source pour n'importe quel objectif
- Permettre la distribution de copie sans paiement de droits de licence :
  - Logiciel original
  - Les copies du logiciel ayant subi une modification mineure ou non au code source.

### Code Source

Tous les logiciels ont été écrit dans ce qui est connu comme étant du code source. Ceci représente les instructions écrites en langage humain par les développeurs de logiciels pour être exécutées par le système. Dans la plupart des cas, ces instructions de programmation sont compilées dans un format nommé « binaire » ou code machine. C'est ce code qui est exécuté sur le système informatique.

La différence majeure entre les logiciels propriétaires et ceux en logiciels libres est que le code source du logiciel libre est disponible à tous. C'est le contraire pour les logiciels propriétaires, dont les codes sources sont interdits d'accès: les clients n'ont accès qu'au code binaire, ou pré-compilé.

Dans la plupart des cas, l'accès au code source ne représente pas un grand avantage, lorsque les utilisateurs ne désirent pas modifier le logiciel, mais seulement l'exécuter. Par contre l'accès au code source réduit le risque de blocage à un seul vendeur.

## Logiciel libre et logiciel commercial

Il est fréquent de considérer le logiciel libre comme étant impossible à commercialiser. Ce qui n'est pas le cas. La plupart des logiciels libres utilisés par les grandes corporations et les gouvernements sont disponibles avec des termes commerciaux. À la différence qu'ils sont disponibles pour une redistribution gratuite.

Les logiciels libres n'excluent pas l'utilisation des logiciels à des fins commerciales. Les logiciels créés en mode libres ont même un droit d'auteur. Par contre le modèle en logiciel libre exige que les utilisateurs aient des droits et des obligations particulières. Un des droits est la redistribution, si une modification est effectuée, les modifications doivent être disponibles à tous.

<b>Logiciels Libres</b>	<b>Logiciels commerciaux</b>
Accessible de différents vendeurs ou emplacement sur Internet	Disponible d'un seul éditeur
Priorité sur les services	Priorité sur les frais de licences.
Fréquente utilisation sur les serveurs et appareils réseaux.	Fréquente utilisation sur les ordinateurs personnels.
Toutes les améliorations apportées par quiconque sur Internet sont intégrées à l'application; ce qui force une évolution rapide du produit.	Les utilisateurs ne peuvent modifier ou améliorer l'application.

## **Modèle de Logiciel Libre**

Il existe une fausse croyance que le « logiciel libre est gratuit et par conséquent une opération de bénévolat des programmeurs, parce que ceux-ci désirent uniquement et simplement contribuer à son développement. »

Ceci est faux. Adam Smith (1723-1790), philosophe du capitalisme, dit dans le livre « The Wealth of Nations » : « Il n'est pas de la bienveillance du boucher, du brasseur, ou du boulanger que nous nous attendons à recevoir notre dîner, mais de leur respect pour leur propre intérêt. Nous ne parlons pas à leur humanité, mais à leur amour d'eux même, nous ne parlons jamais de nos propres besoins mais à seulement de leur avantage » À ce moment, il décrivait un monde dans lequel les intérêts personnels des gens créent des structures mutuellement bénéfiques.

Les contributeurs aux projets en logiciel libres sont attirés pour différentes raisons : en fonction de leurs connaissances techniques, de leurs besoins et des technologies disponibles. Il est possible d'identifier quelques rôles ou chacun sert les besoins de tous; Les intérêts personnels de chacun créent des structures mutuellement bénéfiques:

**Utilisateur** : Il parle de la technologie, il recherche la confiance et le sentiment de sécurité des autres utilisant la même technologie.

**Super utilisateur** : il aide les gens avec leurs problèmes. Il cherche à avoir l'approbation des autres et le sentiment du devoir accompli en aidant les gens.

**Rédacteur, Expert**: Il discute de la technologie dans le forum publique ou article. Il recherche la gloire et la reconnaissance publique puisqu'il est capable d'identifier les tendances du marché.

**Initié** : Il prend une part dans le processus d'assurance qualité du produit. Il cherche à se familiariser avec la technologie, laquelle pourrait devenir importante dans le cadre de son travail ou permettre d'améliorer son curriculum vitae.

**Joueur** : Très actif dans le développement de la technologie, il joue un petit rôle dans la direction du processus. Il recherche la reconnaissance des pairs et la gloire d'être dans une équipe gagnante.

**Joueur Clef** : Il joue un rôle majeur dans le développement et la direction du projet. Il cherche à imposer leurs idées et obtenir une réputation importante. Il veut faire grandir un petit projet en un succès majeur.

**Supporteur** : Il fournis un apport financier au projet, il cherche à vendre le service, souvent aux utilisateurs qui ont besoin de la technologie.

D'autres types de gens existent et sont utiles au projet, tel que les « **trolls** » qui attaquent le projet et ses techniques dans les forums publics. Ceci attire les **rédacteurs** et **experts** à défendre le projet. Toutes ces discussions attisent les ardeurs des gens à tester le projet et à le diffuser; cela augmente la popularité et attire d'autres joueurs aux produits etc.

## **Utilisation des logiciels libres dans l'entreprise**

Toutes les grandes entreprises et organisations publiques utilisent des produits de logiciel libre. Ces logiciels portent ou non l'étiquette de logiciel libre. Les logiciels libres sont beaucoup plus répandus que ce que laissent croire les entreprises..

Voici des exemples d'utilisation de ce type de logiciels libres :

**Infrastructure réseau** : Application de gestion de nom de domaine (DNS), Allocation d'adresse IP (DHCP), service Web, Serveur Proxy, répertoire (LDAP), optimisation des communications, reformation de paquet.

**Serveur de base de donnée** : Les BD très utilisées incluent : Firebird SQL (autrefois Interbase), Ingres, MAXDB (autrefois Adabas), MySQL et PostGreSQL. Également plusieurs base de données propriétaires sont maintenant disponible sur les systèmes d'exploitation en logiciel libre.

**Système de sécurité** : Inclus les coupe-feu, les logiciels de détection d'intrusion, l'analyse du trafic, système de VPN, « Honeypots », logiciel anti-virus et anti-pourriel.

**Publication Internet et Intranet** : Serveur Web, Système de gestion du contenu (CMS), et outils de gestion du flux de travail.

**Gestion documentaire** : Des systèmes de gestion de capture automatique des document électroniques, système de révisions, systèmes d'archive.

**Courriel et communication** : Plusieurs solution de courriel, serveurs de messagerie instantanée, communication partagée. (Calendrier, carnet d'adresse, rappels et répertoires publiques)

**Serveur d'application** : Plusieurs serveurs basés sur PHP, Perl, Python et outils de script ZOPE, Java, Java 2 Enterprise Edition, etc. Plusieurs applications propriétaires fonctionnent maintenant sur les systèmes d'exploitation en logiciel libre.

**Serveur de fichier et d'impression** : Outils disponible pour plusieurs protocoles de partage de fichier tel que Unix NFS, Microsoft SMB/CIFS, Novel netware NCP.

**Stockage** : Plusieurs applications de stockage réseau sont basées sur des plateformes en logiciel libre. (NAS)

**Station de travail limité** : Plusieurs postes de travail ont des fonctions limité, tel que uniquement le courriel, la navigation web ou l'accès à un serveur de terminal. De telles restrictions sont idéales pour des kiosques ou autres utilisation similaires.

**Calcul haute performance** : Système de haute performance de calcul à image simple (Échelle verticale), Image en groupe avec plusieurs systèmes à faible coûts (Cluster ou Échelle horizontale)

**Station de travail haute performance** : Inclus les stations à plusieurs processeur, ceux à 64 bits et à large capacité de mémoire pour les applications exigeantes tel que l'analyse scientifique, la modélisation d'imagerie 3D etc.

## **Survol de l'industrie du Logiciel Libre**

Comme il existe plusieurs différentes plateformes (OS), en logiciel libre, tel que Linux ou les BSD, plusieurs vendeurs d'application indépendante (ISV) ont commencé à produire des versions de leur logiciel afin de supporter ces plateformes. Oracle offre sa célèbre base de données commerciale sur les systèmes d'exploitation Linux. IBM assure une comptabilité entre Linux et ses serveurs etc.

Les grandes entreprises d'impartition ont commencé à supporter ces plateformes et applications. Par contre, le marché des solutions de logiciel libre et ses services est encore dominé par les plus petits joueurs. Il est avantageux pour les entreprises d'avoir des plateformes en logiciel libre puisqu'ils gardent le contrôle sur les fréquences et date des mises à jour. Au contraire, il est fréquent de voir une entreprise avec un système désuet, sans support ou sans autre option qu'effectuer une mise à jour complète du système et forcer le renouvellement de l'équipement informatique.

### ***Préoccupation par rapport au logiciel libre***

Le travail avec le logiciel libre est nouveau pour beaucoup d'entreprises. Ce qui a pour résultat de soulever beaucoup d'inquiétudes et de préoccupations. Les plus fréquentes sont : les coûts de licences, la disponibilité du support, la qualité du produit, sa viabilité compte tenu de sa jeunesse et de son niveau de développement sans structure.

### **Coûts des licences**

La plupart des logiciels libres n'ont pas de frais de licences associées. Les vendeurs génèrent des revenus en fournissant des services de support et de mise en place des technologies. Certains vendeurs fournissent des solutions permettant l'intégration de logiciel commercial avec du logiciel libre. Ce qui est facturé soit par utilisateur ou par serveurs.

Et d'autres vendeurs fournissent des prix en fonction des différents niveaux de support. Par exemple, correctifs de sécurité en 12 heures, résolution de problèmes dans les quatre heures etc. Ce qui permet à l'acheteur d'avoir le degré, le niveau et la vitesse de support approprié à ses objectifs de qualité de service d'entreprise.

### **Disponibilité de support**

Une des questions les plus fréquentes est : Es-t-il possible d'obtenir du support de qualité? Où?

À la base, le niveau de support est fourni par les utilisateurs, répartis dans le monde, et sans niveau minimum de service. Les moyens utilisés vont des chambres de discussions sur IRC, les groupes de discussions par courriel, les documentations disponible sur les sites Internet et finalement les « FAQ » ou foire aux questions des développeurs du logiciel.

Par contre différentes organisations ont décidé de fournir un service de support basé sur le modèle des logiciels commerciaux. Par exemple un frais minimum par mois, ou frais par appel de services.

## **Support interne**

Le modèle en logiciel libre fournit une bonne flexibilité à l'entreprise; Elle peut fournir la responsabilité du support à leur équipe interne de développeur; si elle a les connaissances et les capacités humaines et techniques requises.

## **Support externe**

Certaines entreprises préfèrent offrir le service en impartition, c'est-à-dire à un fournisseur externe. Ce fournisseur doit fournir le niveau 1 et 2 d'assistance technique. Il doit également entreprendre une correspondance et la collaboration nécessaire avec la communauté de développeur du logiciel libre, afin de créer un troisième niveau de résolution d'incident. Du point de vue de l'entreprise, tout ce processus est similaire à celui si l'impartition était fournie à un vendeur de logiciel commercial.

<b>Support Interne</b>	<b>Multiple support externe</b>	<b>Impartition du support</b>
Idéal pour le premier niveau de support.	Réduction du risque liée à l'impartition, compte tenu de la présence de plusieurs fournisseurs.	Incidents automatiquement couverts par les niveaux de services.
Offre de la flexibilité.	Plusieurs types d'options de support en fonction des besoins.	Processus de résolution de problème existant.
	Économie d'échelle	Comptabilité plus simple.

## **Maximiser la flexibilité des options de supports**

Les logiciels libres offrent une très grande flexibilité, due à leurs étapes de développement, de distribution et des types de licences existantes. Les produits en logiciel libre ont une grande variété de type de support fournis par les entreprises locales, que cela soit par appel de service, par services Internet/Courriel ou par assistance de haut niveau.

Ce modèle de support en multipoints est la principale différence entre le logiciel libre et le logiciel commercial traditionnel. La flexibilité des options de support offre aux entreprises la possibilité d'obtenir un meilleur rendement dans les services retenus.

D'un point de vue économique, la proximité des fournisseurs de services permet une meilleure collaboration. Les besoins de l'acheteur sont mieux connus par le fournisseur local de service, et la compétition entre les fournisseurs de produits ou services en logiciel libre permet une grande flexibilité.

## **Fiabilité du logiciel**

Une autre inquiétude relative aux logiciels libres concerne leur fiabilité. Compte tenu que les solutions sont gratuites et qu'aucune entité corporative n'en prend charge, comment est-il possible que cette solution respecte les mêmes normes de qualité qu'une solution propriétaire?

Tout comme pour le logiciel propriétaire, il existe une très grande variété de solutions ouvertes. Les répertoires disponibles sur Internet indiquent environs 100,000 solutions en logiciel libre. Par contre à ce jour, peu d'études comparatives entre le logiciel libre et leur compétiteur propriétaire ont été effectuées.

Les solutions ouvertes font partis de différentes catégories, par exemple, base de donnée, langage de programmation, serveurs d'application etc. Dans chaque catégorie il existe des centaines de solutions, plusieurs de ces produits ne sont pas matures et n'ont pas atteint les standards de qualité requis pour être utilisés dans l'entreprise. Par contre d'autres projets ont fait leur preuve en tant que technologie ou application mature, stable, fonctionnelle et utilisée à grande échelle. Donc afin de démontrer la stabilité du produit, l'historique de la solution doit être comparé et évalué. L'adage qui dit : « Le passé est garant de l'avenir » semble s'appliquer dans cette situation.

## **Maturité et longévité**

Le modèle de logiciel libre ou ouvert est basé sur la collaboration entre les utilisateurs, ce qui a pour effet d'augmenter la maturité et la fiabilité du logiciel au fur et à mesure que le niveau d'utilisation augmente.

La plupart des projets en logiciel libre utilisent le modèle de pensée qui est: « Livrer tôt et livrer souvent ». Les premières versions sont moins fiables et moins robustes puisqu'elles sont habituellement en projets pilotes, afin de voir le degré d'intérêt pour la solution.

Par contre si l'application est programmée et conçue correctement, son développement peut se faire à une très grande vitesse. Inversement, les applications sans communauté fortes devraient être évitées. Il est important de voir que la maturité et la longévité d'une solution en logiciel libre sont basés sur la force, les compétences, l'utilisation diversifiée et la structure portée à la gestion du projet. Les versions de l'application sont « souvent » mises à la disponibilité du public. C'est aux court d'une de ces étapes que des applications possèdent le minimum requis pour que diverses entreprises embarquent dans le train et continuent de pousser son développement, dans leur propre intérêt.

## Scénario de développement

Afin de développer une application, il existe plusieurs possibilités. Les plus courantes décrites ci-dessous représentent les mêmes méthodes que pour les logiciels commerciaux.

- **Développement maison** : Le démarrage, la gestion ou la maintenance du projet est effectué à l'intérieur de l'organisation.
- **Développement externe** : L'embauche d'une ou plusieurs ressources (entreprise) afin de créer, de gérer et de maintenir le projet.
- **Développement accidentel** : Le développement d'une application en logiciel libre se fait dans le cadre d'un autre projet plus gros, le logiciel libre n'est pas la composante principale de ce projet, mais simplement un ajout.
- **Continuité de développement** : L'organisation prend en charge une portion partielle ou complète d'une application en logiciel libre déjà disponible. Attention aux licences d'utilisation si l'organisation compte revendre l'application.

Lors du développement d'application, peu importe le modèle de développement utilisé, les entreprises font face à différents risques qui font grandir les coûts d'exploitation du projet; il est important de tenir compte des facteurs suivants:

- Le niveau d'expertise et de confort des ressources internes par rapport aux technologies utilisées ou désirées.
- La complexité de migration d'une plateforme ou d'une application vers un autre environnement.
- Les coûts et la complexité des changements qui devront être effectués aux données, aux applications, aux protocoles réseau et aux formats des documents.
- Le niveau de formation requis pour utiliser les nouvelles solutions (en cas de roulement de personnel).

## **Développement maison**

« Organisation développe le logiciel par elle-même pour elle-même. »

Cette méthode de développement apporte beaucoup à l'organisation en terme de compétence et de flexibilité. Les gens travaillant sur le développement et la maintenance du logiciel maison en assureront la fiabilité une fois celui-ci implanté à dans l'entreprise. De plus la programmation du logiciel se fait directement sous la gestion du chef de projet qui voit à répondre aux besoins d'affaire exprimés au départ.

Méthode de développement de logiciel maison utilisant les solutions en logiciel libre:

1. Trouver les solutions déjà existantes répondants aux besoins
2. Sélectionner les meilleurs solutions.
3. Évaluer la solution en fonction des critères de rentabilité, voir lequel des logiciels réponds le plus aux besoins d'affaires.
4. Analyser la qualité et la sécurité de la solution
5. Tenter un projet pilote; développement
6. Revoir les résultats du projet pilote vis-à-vis les exigences de bases.
7. Planification pour l'implantation
8. Mise en production du système.

Pour l'entreprise, effectuer du développement maison apporte son lot d'avantages, mais aussi d'inconvénients et de responsabilités. Les besoins exprimés pour effectuer du développement sont nombreux, mais se rapportent généralement aux options suivantes :

- Rendre les processus d'affaire plus simple et efficace
- Diminuer les coûts technologiques
- Simplifier les outils utilisés
- Rationalisation des plate-formes utilisées
- Standardisation des plate-formes utilisées
- Diminuer l'exposition aux risques
- Rendre les plate-formes et applications de l'organisation plus stables
- Réduire l'impact d'une défaillance de système
- Gérer soit même les mises à jours et leur fréquence.

## ***Développement externe ou en impartition***

« Organisation mandate un autre groupe pour la réalisation du travail. »

Lorsqu'une entreprise désire confier à une autre firme la création de logiciel sur mesure, ceux-ci doivent avoir un contrat définissant clairement les attentes de chacune des parties. Définir les exigences minimales du logiciel et du support qui doivent faire partie de ce contrat.

Afin de faire une bonne sélection du fournisseur externe, les considérations suivantes doivent être soulevées :

- Capacité et expérience du fournisseur à offrir le service désiré
- Les niveaux de services fournis, avec une appréciation du service
- Un plan de transition et de gestion de la solution
- Les coûts proposés.
- Respect des processus (affaires et éthiques)
- Qualité de l'équipe du fournisseur
- Profil historique du fournisseur

## ***Développement accidentel***

« Logiciel utile à avoir pour aider la réalisation d'un plus gros projet »

Il arrive que lors du développement d'une application, certains outils manquent aux programmeurs afin de bien réaliser le travail. Par exemple pour tester la puissance de traitement, certain outils tel qu'un simulateur de la demande, pourrait être créé pour tester le projet principal. Une fois le projet terminé, les petits outils créés par les programmeurs sont remis sur Internet en projet personnel par ces mêmes programmeurs.

## ***Continuité de développement***

« S'associer à un projet pour augmenter l'offre de service de l'organisation »

Certain projets en logiciel libre offrent des possibilités très intéressantes pour une entreprise: celle-ci reprends donc le logiciel, continue son développement pour l'intégrer dans un autre projet ou produit afin de le commercialiser.

## **Analyse et gestion du risque**

Pour tout les nouveaux logiciels, qu'ils soient d'acquisition, de développement en logiciel libre ou commerciaux, il existe une phase d'analyse et de gestion du risque.

Le développement et l'utilisation de logiciel libre est nouveau dans le paysage informatique : en plus des différentes inquiétudes liées, ce nouveau mode de développement apporte également de nouveaux risques qu'il est important de gérer afin de prévenir les pertes et problèmes futurs.

### ***Disponibilité des compétiteurs viables***

L'absence de compétiteur représente un risque de haut niveau avec les applications propriétaires, particulièrement dans le cas où un seul fournisseur est disponible. Les conséquences financières pourraient être importantes, dans un cas où le fournisseur ne peut plus fournir les services.

Il existe également une possibilité que le fournisseur bloque l'accès des utilisateurs vers d'autres compétiteurs. Ceci avec des frais importants si le client désire passer d'un fournisseur à un autre. Ceci permet au fournisseur principal de hausser ses tarifs à sa guise et de forcer les niveaux de services qu'il décide, laissant peu de place à la négociation.

Ce scénario n'est pas complètement éliminé avec le modèle en logiciel libre, par contre il est plus facile à éviter, puisque le logiciel libre offre des licences d'utilisation ouvertes et facilite à un plus grand nombre d'entreprise de changer de fournisseur. Ce qui encourage une économie de compétiteur.

Avec le logiciel libre, aucun fournisseur n'a le monopole de la distribution, de la vente et du support des produits. Ce qui a pour conséquence directe d'ouvrir la porte à plusieurs fournisseurs pour des mêmes services.

### ***Garanties et indemnités***

Il est important de ne pas accorder trop de valeur aux garanties offertes par les éditeurs de logiciel commerciaux, puisque ceux-ci offrent une garantie proportionnelle à la valeur d'achat du logiciel et n'engage pas ceux-ci à corriger les erreurs de logiciel.

Les logiciels libre n'offrent généralement aucune garantie aux utilisateurs.

Que le fournisseur soit commercial ou non, il est important pour le client qui désire se procurer un logiciel, d'évaluer le fournisseur en fonction des garanties ou indemnités tel que :

- Capacité à corriger les failles de sécurité
- Maintenance des problèmes d'applications
- Assistance technique
- Gestion des situations problématiques

## Cycle de vie du logiciel

Une des différences majeures entre les logiciels libres et commerciaux réside dans leur modèle de développement. Les projets en logiciels libres, typiquement rendent disponibles les versions des logiciels très rapidement et offrent un cycle de développement très décentralisé.

En terme général, les logiciels libres sont produits et livrés souvent et rapidement: aussitôt qu'une fonctionnalité est disponible, le produit est rendu public. Ceci explique pourquoi bon nombre de produits ne sont pas matures et ni fiables à leur première distribution.

### Première phase : Développement de la solution

C'est à cette phase que le produit est initialement créé et mis à la disposition de tous. L'organisation du projet se crée et se structure.

- Période de changement technologique rapide
- Logiciel disponible à tous, mais peu ont les capacités techniques pour le tester.
- L'installation du logiciel est très complexe.
- De nouvelles versions du logiciel sortent chaque semaine ou plus souvent encore.
- La documentation est minimale
- Peu d'utilisateurs du logiciel
- Le support du logiciel se fait uniquement par les groupes de discussions
- Le logiciel manque plusieurs des fonctionnalités importantes et n'est pas raffiné.

En contradiction les fournisseurs de logiciels commerciaux n'offrent pas leur produit dans ces versions primaires. Ils attendent généralement que toutes les fonctions soient disponibles avant de permettre à leur client de tester le produit. Et même lorsque le produit est disponible en version « beta test » il est habituellement beaucoup plus près de la fin que du début du développement.

### Deuxième phase: Première adoption

Cette phase commence habituellement par un ralentissement du développement, lorsque le produit est plus stable, toutes les fonctionnalités sont disponibles et documentées. De plus en plus d'utilisateurs adoptent le produit. Les utilisateurs commencent habituellement un usage en zone de production à cette étape.

Les attributs des logiciels dans cette phase sont:

- Une version 1.0 est généralement fiable
- La technologie est mature, avec une installation simplifiée et mieux documentée.
- Des fournisseurs commerciaux commencent à offrir des niveaux de support.

Cette phase peut durer plusieurs années et donner des logiciels très fiables.

## **Phase long terme ou de maintenance**

Le produit a accumulé toutes les fonctionnalités que les utilisateurs désirent, il ne reste qu'à maintenir et résoudre les problèmes de sécurité ou « bugs » découverts.

Les attributs des applications à cette étape sont :

- Stabilité et maturité du logiciel dans tous ses aspects
- La confiance dans le produit est le résultat de plusieurs années de performance et de test.
- La plupart des failles de sécurité et des problèmes ont été découverts et corrigés.
- Les utilisateurs ne désirent pas qu'il soit modifié, mais simplement mis à jour.

## **Licences d'utilisation**

La différence majeure entre les logiciels propriétaires et libres réside dans les licences d'utilisation. La plupart des licences de logiciels propriétaires contiennent des termes similaires, donc les utilisateurs ont maintenant une connaissance beaucoup plus grande. Ces licences se concentrent sur ce que les utilisateurs peuvent faire ou ne pas faire avec les logiciels, comme le nombre d'utilisateurs par licence, ou le nombre de systèmes utilisant le logiciel.

Dans le cas des logiciels libres, les licences se concentrent d'avantage sur la redistribution du logiciel et l'accès aux codes source.

Comme résultat, les risques inhérents à l'acquisition de logiciel libre sont différents de ceux des logiciels propriétaires.

Il existe plusieurs dizaines de type de licences « libres » en ce moment. Certaines sont plus restrictives que d'autres concernant l'utilisation commerciale que l'utilisateur peut en faire.

Les deux licences les plus utilisées sont :

- GPL: GNU General Public Licence
- BSD: Berkeley Software Distribution

Par contre, beaucoup d'autres types de licences existent puisque tous les créateurs de logiciel peuvent appliquer les restrictions qu'ils désirent à leur création.

### **GNU General Public Licence (GPL)**

Cette licence est la plus répandue, elle assure que toutes les versions du logiciel découlant du logiciel original ont la même licence. C'est à dire que tous les logiciels qui utilisent du code source d'un logiciel protégé par GPL doivent utiliser la licence GPL.

La licence stipule que toute modification effectuée au code source et redistribuée doit être faite dans un tout. C'est à dire que les nouveaux utilisateurs qui commencent à utiliser le nouveau logiciel doivent avoir accès, sans frais, au code source dans son intégralité et eux aussi doivent avoir la possibilité de faire des modifications et de le redistribuer par la suite.

Cette licence permet à tous les utilisateurs et programmeurs de modifier le logiciel et ainsi de permettre une évolution rapide du produit dont tous peuvent profiter, sans frais.

### **Berkeley Software Distribution (BSD)**

La licence BSD est la plus permissive de toutes, cette licence permet à une entreprise de prendre le logiciel, de le modifier et de revendre ce nouveau logiciel sans avoir à fournir le code source de leur modification.

Cette licence est souvent utilisée dans les appareils réseau populaires comme les Linksys.

## Référence en logiciel libre

Licences :

<http://www.gnu.org/philosophy/philosophy.html#LicensingFreeSoftware>

GNU General Public licence :

<http://www.gnu.org/copyleft/gpl.html>

Répertoire des licences en logiciel libre :

<http://www.opensource.org/licenses/index.php>

Free software foundation :

<http://www.fsf.org/>

Sites de références de logiciels libres disponibles :

<http://sourceforge.net>

<http://freshmeat.net/>

Site suivant la progression de développement des distributions des systèmes d'exploitation :

<http://distrowatch.com/>

## Bibliographie

The risk of closed source computing :

[http://www.ibiblio.org/oswg/oswg-nightly/oswg/en\\_US.ISO\\_8859-1/articles/alan-cox/risks/risks-closed-source/risks.html](http://www.ibiblio.org/oswg/oswg-nightly/oswg/en_US.ISO_8859-1/articles/alan-cox/risks/risks-closed-source/risks.html)

OpenSource Initiative :

<http://www.opensource.org/>