Avantages d'une coexistence entre logiciels "open source" et commercial

Tuukka Turunen, Directeur Recherche et Développement chez Qt, et Mirko Boehm, conseiller économique et ex-président de KDE e.V. expliquent comment les modèles commerciaux peuvent renforcer les communautés de logiciel open source pour apporter de réels bénéfices aux concepteurs système

Dans le monde du développement logiciel, de nombreux débats sur les avantages de l'open source (logiciel libre dont le code source est disponible) par rapport au logiciel commercial continuent d'avoir lieu. Les développeurs, les éditeurs de logiciels, les utilisateurs et les acheteurs mettent tous en avant leurs arguments respectifs, souvent pour protéger leurs intérêts acquis.Cependant, comme souvent dans la vie, la réalité n'est ni toute noire ni toute blanche, mais elle est faite de nombreuses nuances de gris. Différents modèles et dispositions se traduisent par le fait que le logiciel open source peut être disponible sans licence payante, ou en tant que composant d'une offre commerciale plus large associée à une licence séparée. Mais ces deux approches ne sont pas mutuellement exclusives. Dans beaucoup de cas, non seulement des offres commerciale et open source peuvent coexister sans problème, mais cela renforce réellement l'offre produit et la communauté vers laquelle le concepteur peut se tourner pour obtenir du support.

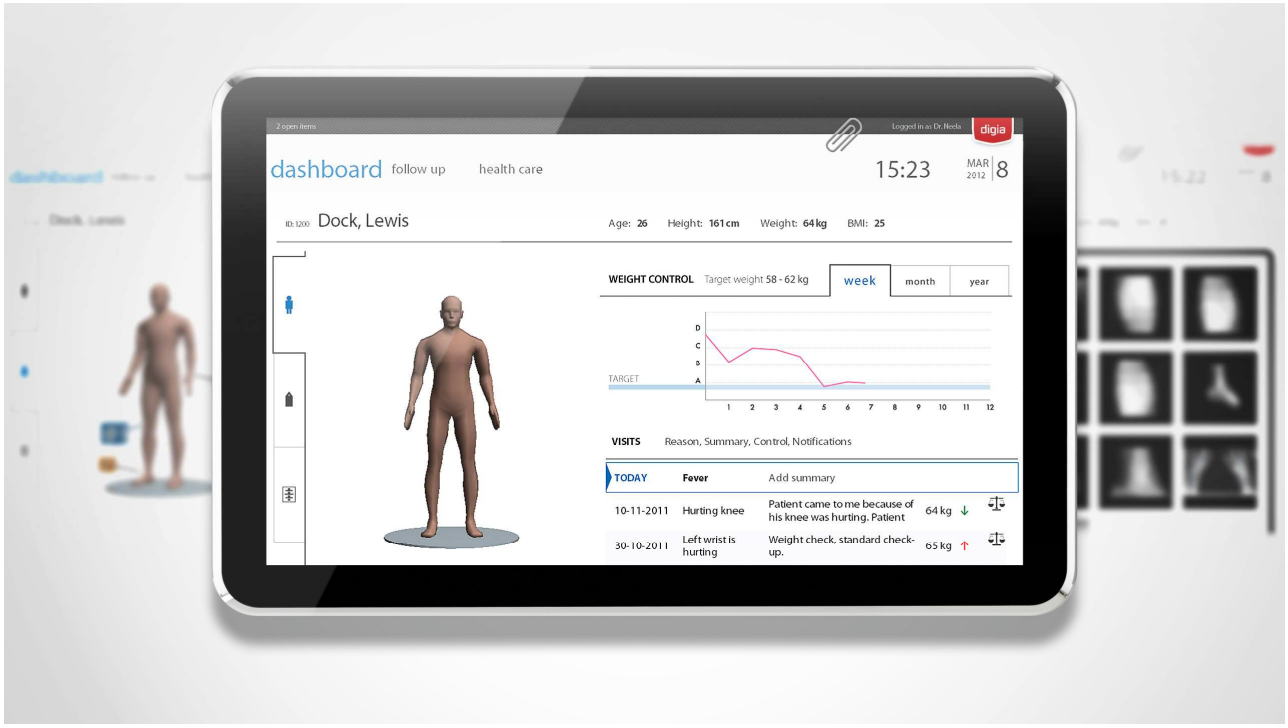
Croissance de l'Open Source
Depuis son apparition et sa promotion sous la bannière du "mouvement pour le logiciel libre" dans les années 80, l'utilisation de logiciels open source (le nom plus "politiquement correct" sous lequel on le désigne depuis les années 90) s'est développée rapidement. Typiquement couvert par des licences autorisant l'inspection, la modification, l'amélioration et la distribution du code source, le logiciel open source (OSS pour Open Source Software) se rencontre désormais dans des applications allant des machines industrielles aux équipements médicaux, et de l'électronique grand public aux systèmes de défense.



Parmi les divers avantages avancés pour utiliser des OSS, on trouve le coût (les logiciels open source sont en général gratuits), la souplesse de déploiement, et le fait de ne pas dépendre d'un fournisseur unique. Les contre-arguments des partisans des logiciels commerciaux sont également le coût (compte tenu du fait qu'investir dans de l'expertise technique ou du support technique à long terme signifie que déployer un OSS n'est jamais réellement gratuit). En l'absence d'effort de développement et de support commercial une entreprise est susceptible de devoir maintenir le produit OSS elle-même, ce qui peut s'avérer plus coûteux à long terme qu'une approche propriétaire classique. Le manque de documentation et les risques de protection de copyright et de marque commerciale associés aux logiciels open source sont aussi souvent cités.
Néanmoins, pour la plupart des défenseurs des OSS, le premier avantage est de loin le support de communautés (souvent passionnées) dont le but est de collaborer pour améliorer sans cesse la qualité, les performances, la fiabilité et la flexibilité du produit qu'elles ont choisi. Ces communautés non seulement servent de conseillers techniques pour le logiciel, mais contribuent aussi à l'enrichissement du code, agissent comme des ambassadeurs de la marque, et entrainent en général les jeunes talents à créer de nouveaux produits et de nouvelles technologies.
Notre expérience montre que les éditeurs commerciaux et les développeurs open-source travaillent en symbiose et permettent aux utilisateurs de profiter du meilleur des deux mondes, pour peu que les deux parties soient engagées dans une relation de long terme, et soient ouverts quant à leurs objectifs.
Les entreprises peuvent contribuer aux projets logiciels open-source, tout comme des volontaires, mais leurs contributions sont en général d'une nature différente. Les contributions nécessaires aux projets peuvent généralement être classées en trois catégories : temps, argent et expertise. Il est plus facile aux entreprises de contribuer financièrement et sous forme d'expertise autre que les volontaires. Plus ces contributions sont complémentaires de celles des volontaires, plus le logiciel open source bénéficie de ce genre de coopération.
Une discussion de ce type ne serait pas complète sans mentionner au moins brièvement les

différentes options de licence logicielle proposées aux développeurs. Le véritable logiciel open source, par exemple, est généralement régi par une licence GPL (General Public Licence, ou licence grand public) selon les termes de laquelle le code source et les modifications doivent être partagés avec l'utilisateur final. Le logiciel peut être vendu mais est en général disponible sans frais de licence. Le support (si nécessaire) est acheté séparément et la diffusion ne donne pas lieu au paiement de droits (aussi appelés "royalties"). La société Canonical, par exemple, mène une activité de ce type en supportant la distribution GNU/Linux Ubuntu. A l'inverse, une licence de logiciel commercial prévoit typiquement le paiement de droits et ne comporte aucune obligation de partager le code source. Par conséquent, cela permet à des utilisateurs de développer des applications propriétaires et de facturer des licences d'exécution (dites licences "runtime"). Les licences commerciales sont également presque toujours associées à un certain niveau de support.
Il existe également une troisième catégorie de licence connue sous le nom de LGPL (Lesser General Public License ou Licence Générale Publique Limitée). Elle ressemble à la licence GPL dans la mesure où le code source du logiciel et de tous ses dérivés doit être accessible au public, mais avec la distinction subtile que le logiciel peut être utilisé pour créer des applications propriétaires.

Entreprises commerciales dans l'écosystème Open Source
En contribuant financièrement au développement jusqu'alors collaboratif d'un produit, les entreprises définissent en quelque sorte leur propre rôle au sein des communautés open source. Elles font cela d'une part en assumant une position de membre à part entière de la communauté autour du produit, et d'autre part en choisissant ce sur quoi elles travaillent. En fonction de l'engagement à plus ou moins long terme d'une entreprise vis-à-vis de la communauté, son rôle de contributeur peut prendre différentes formes.
Premièrement, les entreprises restent concentrées sur les sujets commerciaux, c'est-à-dire qu'elles se préoccupent du client. La participation d'entreprises dans les communautés open source conduit généralement à un produit plus achevé, avec une expérience utilisateur plus éprouvée.
Deuxièmement, les entreprises assurent davantage de stabilité et de continuité au processus de développement des produits. En affectant des équipes de manière permanente au projet, les fluctuations de contributeurs volontaires se trouvent lissées, et un effort plus soutenu peut être produit. L'inconvénient de ceci est que les entreprises ont généralement besoin d'une organisation plus formelle. Cela peut provoquer des conflits avec l'organisation personnelle des contributeurs non-commerciaux, aussi les entreprises doivent elles veiller à éviter la désertion des contributeurs volontaires.
Troisièmement, les contributeurs open source volontaires motivés s'arrogent naturellement certaines tâches en fonction de leur intérêt personnel, et en général laissent de côté les tâches les plus courantes, qui peuvent pourtant offrir un grand intérêt pour les utilisateurs, comme par exemple la correction de bugs. Les entreprises peuvent affecter des équipes qualité à certains produits, pour épauler les contributeurs volontaires de la communauté.
Somme toute, le rôle des entreprises au sein des communautés open source sera bénéfique, pour peu qu'une bonne collaboration existe entre les groupes volontaires et les groupes commerciaux.



Collaboration entre commercial et Open Source

Une bonne illustration de la façon dont la disponibilité d'un logiciel à la fois en version commerciale et en version open source peut bénéficier à la communauté, est Qt, un environnement regroupant une bibliothèque de classes modulaire C++ et des outils de développement. Qt est largement utilisé, aussi bien pour développer des applications multiplateformes nécessitant une interface utilisateur graphique (GUI), que pour des programmes sans GUI, comme des outils à ligne de commande ou des consoles pour serveurs. L'environnement Qt, qui intègre des API de programmation C++ et de style CSS/JavaScript, un IDE (environnement de développement intégré), Qt Creator, et un certain nombre d'outils et de suite d'outils, permet aux développeurs d'écrire une application une fois, puis de la déployer sur de multiples matériels, tournant sous différents systèmes d'exploitation.

En 2012, Digia a acquis l'ensemble des technologies logicielles et des activités Qt de Nokia. La plateforme Qt originale a été développée par la société norvégienne Trolltech puis, en 2008, a été acquise par Nokia. Les opérations de service professionnel et de distribution commerciale Qt ont été revendues à Digia en 2011, année qui a vu naître la communauté Qt Project. Qt Project veille à ce que le développement de Qt soit régi comme un véritable projet open source, au sein duquel les développeurs indépendants comme les développeurs de Digia contribuent à l'évolution de Qt.

Naturellement la question évidente est pourquoi, alors que Qt est gratuit, un développeur voudrait payer une licence commerciale ? Clairement le choix dépendra dans une certaine mesure de l'application cible, mais il existe un certain nombre de raisons pour opter en faveur de l'option commerciale, car au bout du compte il se peut que celle-ci s'avère la plus économique pour l'entreprise utilisatrice de Qt. La licence LGPL, par exemple, comporte certaines restrictions quant à la capacité des utilisateurs à reconnecter les bibliothèques et d'autres restrictions susceptibles d'engendrer des contraintes d'architecture qui impacteront les coûts de déploiement et d'entretien pour les entreprises. La licence Qt commerciale protège également l'IP en supportant le développement de logiciel 100% propriétaire, tout en offrant un bon niveau de support dans le cadre de la licence payante.

En outre, du fait des restrictions à prendre en compte si l'on démarre un développement avec une licence open source, puis en optant ultérieurement pour une version commerciale, il peut s'avérer payant d'acheter une licence commerciale dès le début du projet. Le développeur a alors toute liberté pour décider du type de licence (commerciale ou open source) applicable au produit final lors de sa distribution. Pour cela, il leur suffit de créer une application reliée dynamiquement aux

bibliothèques Qt, puis lors de la distribution, de décider de lier soit aux bibliothèques Qt à licence commerciale soit aux bibliothèques open source.

Comment les logiciels commerciaux et open source se renforcent mutuellement

Indépendamment du type de licence choisie, les développeurs individuels comme les concepteurs système et l'ensemble de la communauté tirent profit de la présence de l'offre commerciale, et inversement.

Prenons par exemple, la question des bugs, éternel défi pour un logiciel en constante évolution. Des centaines d'employés de Qt travaillent à l'identification de bugs existants ou potentiels et créent des correctifs pour les résoudre. Ces individus ne sont pas nécessairement plus doués que les autres développeurs de la communauté Qt, mais ce sont des gens ayant une bonne expertise de Qt qui consacrent presque tout leur temps à tester et utiliser la plateforme, et qui sont payés pour obtenir des résultats. Pour autant, les correctifs qu'ils élaborent sont mis gratuitement à la disposition de Qt Project.

Au sein de la communauté Qt, il existe des centaines de milliers d'utilisateurs et des milliers de contributeurs actifs utilisant les toutes dernières versions de Qt pour développer leurs applications. Ces personnes participent à l'amélioration de Qt en signalant les problèmes qu'ils rencontrent. Beaucoup contribuent aux correctifs qui enrichissent Qt Project. Grâce à la gouvernance ouverte, de nombreuses nouvelles fonctions trouvent leur chemin dans Qt à partir des entreprises ou des communautés open source.

Enfin se pose le problème critique du support. Les utilisateurs de versions open source doivent payer pour accéder au support en cours de développement ou au cours de la vie du produit. Une entreprise comme Digia, qui tire ses revenus et investit dans le support à ses clients ayant acheté des licences commerciales, peut aussi mettre à disposition un niveau de service similaire en proposant du conseil payant aux entreprises ayant opté pour la voie open source. L'objectif est d'assurer le succès des clients, indépendamment du type de licence choisi.

Conclusion

Il est clair à travers l'exemple de Qt que lorsque les logiciels commerciaux et open-source coexistent, la communauté dans son ensemble peut en tirer des bénéfices importants, aussi bien en termes d'évolution que d'accès au support, ce qui permet d'accélérer et de faciliter le développement d'applications. Pour une entreprise qui propose des licences commerciales, être membre actif d'une communauté prend tout son sens, non seulement pour l'audience que cela apporte à ses produits, mais aussi pour le retour d'expérience et les contributions d'autres membres, qui permettent au bout du compte d'améliorer les produits et les services que l'entreprise propose à ses clients commerciaux.

A propos de Digia et Qt

Digia est en charge de toute l'activité Qt, y compris du développement produits et de la distribution commerciale et open source. Depuis mars 2011, Digia Plc, entreprise de services et de conseil basée à Helsinki et cotée au NASDAQ OMX (DIG1V), est le distributeur commercial de Qt. En septembre 2012 Digia a racheté à Nokia l'ensemble de l'activité Qt. Digia opère en Chine, en Finlande, en Norvège, en Russie, en Suède et aux Etats-Unis.

Digia dispose d'une grande expertise Qt et s'est forgé une solide expérience grâce au développement de nombreux projets complexes et au travail de centaines de développeurs certifiés Qt en interne. Nous avons aidé plusieurs grands acteurs mondiaux à déployer des projets Qt sous des plateformes desktop, embarquées et mobiles.

L'objectif de Digia est de fournir aux développeurs et aux entreprises des secteurs du desktop, des systèmes embarqués, et du mobile, l'interface utilisateur et l'environnement applicatif multiplateformes les plus puissants. En plus de ses capacités de distribution, de support et de services, Digia Qt a pour mission de collaborer étroitement avec les développeurs pour garantir que leurs projets soient déployés dans les temps, sans dépassement de budget et avec un avantage

compétitif.

A propos de KDE

KDE est une équipe internationale coopérant au développement et à la distribution de logiciel gratuit, open source, pour l'informatique de bureau et l'informatique mobile. Notre communauté a développé un grand nombre d'applications pour les communications, le travail, l'éducation et le divertissement. Nous nous efforçons plus particulièrement de trouver des solutions innovantes à des problèmes anciens ou nouveaux, en créant un contexte dynamique propice à l'expérimentation. Pour les utilisateurs sous Linux ou Unix, KDE propose une suite complète d'applications d'espace de travail utilisateur, qui permettent l'interaction avec ces systèmes d'exploitation au moyen d'une interface utilisateur graphique et moderne. Il s'agit notamment de Plasma Desktop, l'interface bureau innovante de KDE. D'autres applications d'espace de travail sont incluses pour faciliter la configuration système, l'exécution de programmes, ou l'interaction avec des périphériques matériels. Si les espaces de travail KDE tout intégrés ne sont disponibles que sous Linux et Unix, certaines de ces fonctions sont disponibles pour d'autres plateformes. KDE s'appuie sur Qt. Digia est un Supporting Member (membre actif) de KDE e.V.