

Coordonner les flux de contributions (Libres Conseils 30/42)

Chaque jeudi à 21h, rendez-vous sur [le framapad de traduction](#), le travail collaboratif sera ensuite publié ici même.

Traduction Framasoft : [merlin8282](#), [Sphinx](#), [Julius22](#), [goofy](#), [Corentin](#), [lerouge](#), [Asta](#), [peupleLà](#), [okram](#), [Alpha](#), [lamessen](#)

Au confluent de l'amont et de l'aval

Vincent Untz

Vincent Untz est un activiste passionné du logiciel libre, un amoureux partisan de GNOME, ainsi qu'un élément moteur d'[openSUSE](#). Il a été responsable des versions de GNOME de 2008 à 2011, jusqu'à la sortie de GNOME 3.0 ; il a été directeur exécutif de la fondation GNOME (2006-2010) et il dirige l'équipe GNOME chez openSUSE. Quoi qu'il en soit, il trouve plus simple de se décrire comme un « touche-à-tout » (NdT : en français dans le texte) et il travaille dans divers services (certains diraient au petit bonheur la chance) du bureau pour aider openSUSE à rester extraordinaire. Vincent continue à faire du forcing pour que le français soit la langue officielle de GNOME et espère bien réussir bientôt. Sinon, il aime la crème glacée.

Il y a bien longtemps, dans une

chambre, la nuit...

J'étais en train de jeter un dernier coup d'œil sur une liste de bogues pour voir si je n'avais pas oublié de fusionner un correctif. Je m'étais bien assuré d'écrire ce que je pensais être une entrée NOUVEAUTÉS au sujet de la nouvelle version. J'ai entré `make distcheck` (NdT : commande GNU permettant de créer un paquet et de le tester automatiquement dans un répertoire différent de celui de développement pour démarrer le processus de diffusion) et je regardais la console afficher des centaines de lignes. Une archive avait été créée, et j'ai à nouveau vérifié que l'archive se construisait correctement. J'ai vérifié, encore et encore : j'étais inquiet. D'une certaine manière, je ne faisais pas totalement confiance à la commande `make distcheck`. Après avoir tout vérifié plusieurs fois, j'ai envoyé l'archive sur le serveur et expédié un courriel d'annonce.

J'avais réussi à le faire : j'avais sorti ma première archive d'un logiciel sur le développement dont j'étais récemment devenu co-responsable. Et j'ai certainement pensé : « Ah, maintenant les utilisateurs vont pouvoir apprécier un bon truc ! ». Mais à peine quelques secondes après le chargement de mon archive, quelques personnes l'ont téléchargée et ont rendu ma version réellement accessible aux utilisateurs.

C'est une chose que je tenais pour acquise, car je pensais que c'était une tâche banale. J'avais tort.

Amont et aval

De nombreuses personnes participent au processus d'acheminement du logiciel. Et cet effort se répartit généralement entre deux groupes de personnes d'égale importance dans la manière dont fonctionne le logiciel libre aujourd'hui.

En amont : c'est le groupe qui crée le logiciel. Il inclut évidemment les programmeurs mais, en fonction du projet, d'autres catégories de contributeurs sont également essentielles : designers, traducteurs, rédacteurs de documentation, testeurs, trieurs de bogues, etc. En général, l'amont se charge seulement de livrer le code source sous forme d'archive.

En aval : c'est le groupe responsable de la distribution du logiciel aux utilisateurs. Tout comme en amont, les contributeurs ont une gamme de profils très variée et travaillent à la traduction, la documentation, les tests, le triage de bogues, etc. Il y a cependant un profil qui, jusqu'à présent, est spécifique au travail en aval : le packager, qui prépare le logiciel afin de le rendre disponible sous forme de paquet, un format mieux adapté à un usage facile que le seul code source.

Chose intéressante, les utilisateurs ont plutôt bien l'intuition de cette séparation également, bien que nous n'en soyons pas conscients : nous supposons souvent que les développeurs de logiciels sont inaccessibles et nous envoyons plutôt nos retours et demandes d'aide aux distributeurs.

Pour éclairer cette séparation entre amont et aval, une analogie parlante et classique est celle du circuit des biens de consommation, avec les magasins de détail (« l'aval ») qui distribuent des produits manufacturés (« l'amont ») et jouent un rôle important pour les clients (« les utilisateurs »).

Un regard plus attentif sur l'aval

Si je devais résumer le rôle de l'aval en une phrase, voici comment je le décrirais :

L'aval est le pont entre les utilisateurs et l'amont.

Lorsque j'ai sorti ma première archive en amont, je supposais

que, pour l'aval, le travail consisterait principalement à compiler la source pour construire un paquet avec, et rien d'autre. Construire un paquet est effectivement la première étape. Mais c'est seulement le début du voyage vers l'aval : différentes tâches viennent ensuite. Certaines sont purement techniques tandis que d'autres sont sociales. Je vais me contenter de décrire très brièvement ce voyage ici, de manière non-exhaustive, puisque ça pourrait faire l'objet d'un chapitre entier de ce livre (1).

La construction du paquet proprement dit peut se révéler moins triviale que prévu. Il n'est pas rare qu'un *packager* rencontre des problèmes qui étaient inconnus de l'amont. Comme lorsqu'une nouvelle version du compilateur est utilisée (avec de nouvelles erreurs), qu'une bibliothèque spécifique a d'abord besoin d'être mise à jour (parce que l'archive utilise de nouvelles interfaces de programmation) ou bien que le système de compilation de l'archive est conçu pour une certaine façon de fonctionner (qui ne suit pas les directives de la distribution cible). Ce qui est encore plus méconnu par beaucoup est le fait que tous ces problèmes peuvent également se produire après que l'archive a déjà été *empaquetée*, comme lors de la migration d'une distribution entière vers un nouveau compilateur ou bien une nouvelle chaîne de compilation. Aucun de ces problèmes techniques n'est vraiment compliqué à résoudre en lui-même, et l'amont est souvent content de contribuer à la solution. Mais sans l'aval, ces problèmes pourraient ne pas être remarqués par l'amont avant un long moment.

Ce qui selon moi est plus important que ces défis techniques, c'est que l'aval est généralement en contact avec davantage d'utilisateurs que l'amont. Cela se traduit par des rapports de bogue, des demandes de support, des requêtes de changement de la configuration par défaut ou bien d'autres choses encore. C'est là que la foule en aval donne la mesure éclatante de sa force : au lieu de simplement transférer ça en amont, l'aval

va travailler sur les retours des utilisateurs afin de ne renvoyer que des synthèses qui seront utilisables en amont. Bien souvent, les rapports de bogue sont soumis avec trop peu d'informations sur le problème (auquel cas l'aval demandera plus de détails). Souvent, les demandes de support sont issues d'une incompréhension du côté de l'utilisateur (ce que l'aval peut, parfois, traduire par une suggestion visant à modifier le programme afin d'éviter cette incompréhension). Souvent, de nouveaux paramètres par défaut sont suggérés sans réflexion suffisante (l'aval travaillant alors avec les utilisateurs pour voir si le raisonnement est valide). À partir de cette énorme quantité de données, l'aval produira un ensemble d'informations plus compact que l'amont sera en mesure de digérer facilement. Ce qui amènera à des améliorations sur le logiciel.

Il existe généralement deux récompenses pour les contributeurs en aval : les contributions directes et indirectes vers le projet en amont grâce aux efforts effectués par l'aval sont suffisantes pour beaucoup. Mais plus important encore, le contact direct avec davantage d'utilisateurs amène à recueillir la satisfaction qu'ils expriment. Et une situation aussi gratifiante rend facilement heureux beaucoup de gens.

Une petite note au passage : lorsqu'on considère la quantité de travail fournie en aval, je ne serais pas surpris qu'au bout du compte, beaucoup de contributeurs en amont soient bien contents d'avoir des gens agissant comme intermédiaires : cela diminue significativement la quantité de retours tout en améliorant leur qualité (en évitant les commentaires en double, les problèmes non documentés, etc.). Cela permet à ceux qui travaillent en amont de rester focalisés sur le développement lui-même, au lieu de les obliger à soit trier les retours, soit les ignorer.

Rien qu'en regardant mon expérience en amont, je ne compte plus le nombre de correctifs que j'ai reçus de l'aval pour résoudre des problèmes de compilation. Je me rappelle aussi

d'innombrables discussions que j'ai eues à propos des bogues qui affectaient le plus les utilisateurs et qui m'ont permis de prioriser mon travail. De fait, depuis que j'ai rejoint les équipes en aval, j'ai commencé à faire remonter des correctifs proches de ceux liés à des problèmes de compilation à l'amont et à discuter avec ma base en aval pour transmettre des retours d'expérience d'utilisateurs. Une telle collaboration amont-aval participe à l'amélioration de la qualité générale de notre écosystème du logiciel libre et je la considère comme essentielle à notre bonne santé.

Remonter de l'aval vers l'amont !

Je crois fermement que, pour qu'un projet réussisse, il faut qu'il y ait une forte collaboration amont-aval. Je doute que beaucoup désapprouvent. Cependant, par « aval », les gens pensent généralement au travail fait dans les distributions. Mais, particulièrement pour les applications, il devient de plus en plus viable de pousser ce travail fait en aval en dehors des distributions et de tirer parti d'un tel mouvement vers l'amont.

Des outils comme l'[Open Build Service](#) (NdT : distribution *open source* dédiée à la compilation de paquets pour diverses distributions GNU/Linux) permettent plus facilement d'avoir des personnes qui compilent et distribuent des paquets d'une application pour plusieurs distributions. Cela présente des avantages à la fois pour les utilisateurs (qui peuvent profiter plus rapidement et plus facilement des mises à jour de leurs applications préférées) et pour l'amont (qui peut aider à construire une relation plus forte avec sa base d'utilisateurs). Le seul défi qu'un tel mouvement représente est le besoin perpétuel d'avoir quelqu'un qui s'occupe de l'empaquetage, mais aussi qui gère des retours plus nombreux des utilisateurs. Dans les faits, il y a toujours besoin de quelqu'un pour faire le travail de l'aval, sauf qu'il serait fait au sein de la branche amont.

Pour moi, cela semble une perspective excitante et j'irais même plus loin en suggérant que nous, la communauté du logiciel libre, devrions migrer lentement le travail d'aval fait sur les distributions vers un travail d'aval fait directement, aussi souvent que possible, en amont. C'est souvent possible, au moins pour les applications. Cela requiert évidemment de penser différemment. Mais ça permettrait de partager un travail qui, actuellement, est le plus souvent dupliqué sur toutes les différentes branches en aval.

Pour ceux qui souhaitent actuellement commencer à contribuer aux applications qu'ils apprécient, ce travail sur les paquets en amont est une toute nouvelle approche qui pourrait vraiment être une réussite !

J'ai essayé, je suis resté. Pourquoi pas vous ?

L'aval a toujours été essentiel à ma vie en tant qu'utilisateur de logiciels libres – après tout, seules quelques personnes installent manuellement leur système à partir de zéro et je n'en fais pas partie. Cependant, c'est aussi devenu un atout pour moi en tant que développeur en amont, étant donné que j'ai commencé à prendre plus de temps pour discuter avec des personnes en aval afin d'obtenir plus de retours sur les bogues, les fonctionnalités, la qualité générale et même les directions futures du logiciel sur lequel je travaillais.

C'est seulement lorsque j'ai commencé à être moi-même en aval que j'ai compris que cette position est en effet privilégiée afin de conseiller en amont, grâce au contact direct avec les utilisateurs et la perspective différente que l'on retient de cette position différente.

Sans l'aval, nous ne serions pas là où nous sommes

aujourd'hui. Si vous souhaitez avoir un impact significatif, soyez persuadé qu'en participant en aval et en discutant avec l'amont, vous réussirez.

Et vous y prendrez du plaisir.

(1) Note de l'auteur : Il est bon de mentionner que je ne crois pas que l'aval devrait modifier significativement le logiciel mis à disposition par l'amont. Certains, en aval, le font tout de même et cela s'ajoute à leur charge de travail.