



Développement de logiciels libres

Chapitre 1 Contribuer à un projet libre



Pablo Rauzy <pr@up8.edu>
pablo.rauzy.name/teaching/dll

Contribuer à un projet libre

- ▶ S'inscrire aux listes de diffusions.
- ▶ Être présent·e sur les canaux IRC / Discord / Slack / Mattermost.
- ▶ Tester les version beta et RC.
- ▶ Faire des rapports de bug (et de confirmation).
- ▶ Traduire le projet dans d'autres langues.
- ▶ Écrire de la documentation / des tutoriaux.
- ▶ Améliorer la communication (site web, artwork).
- ▶ Dons (argent, machines, ...).

Liste de diffusion

- ▶ Souvent plusieurs listes : utilisateurs / développeurs.
- ▶ Répondre aux questions d'autres utilisateurs et utilisatrices.
- ▶ Interagir avec les développeurs.

- ▶ Comme pour les listes de diffusion mais souvent un seul canal / projet.
- ▶ Discussions plus directes.
- ▶ Premiers contacts.
- ▶ Élaborations collectives d'idées ensuite transmises sur les listes de diffusions ou sur les bugs/issues trackers.

- ▶ Tester les nouvelles fonctionnalités dans votre environnement utilisateurs.
- ▶ Se signaler comme volontaire pour tester les versions beta et release candidate.
- ▶ Faire des rapports sur ce qui va et ne va pas.

Rapports et confirmation de bug

- ▶ Quand il y a un soucis, faire un rapport de bug (IRC → issue tracker).
- ▶ Le rapport de bug doit être précis et utile :
 - détailler l'environnement (OS, versions des logiciels et bibliothèques utilisées),
 - lister les étapes permettant de reproduire le bug,
 - si possible fournir un MWE (*minimal working example*).
- ▶ Il est aussi possible de faire des *feature requests* :
 - détailler la pertinence,
 - expliquer le cas d'usage.

- ▶ Souvent dans des fichiers séparés (.po).
- ▶ Parfois faisable au travers d'interface web pour simplifier la contribution.
- ▶ Pas seulement pour les nouvelles langues :
 - amélioration / correction des langues existantes,
 - traduction / adaptation des nouveaux éléments de l'interface et de la documentation.

- ▶ Écrire de la documentation orientée utilisateur.
- ▶ Souvent pas une priorité quand le projet est portée par une petite équipe.
- ▶ Intéressants aussi : tutoriaux / blogposts.

- ▶ Promouvoir le projet.

- ▶ Toutes les choses qu'on a dites pour les utilisatrice·s restent vraies.
- ▶ Écrire des tests.
- ▶ Supprimer des warnings.
- ▶ Ajouter des commentaires.
- ▶ Corriger un bug.

- ▶ Ajout de tests automatisés de non-régression (pour des fonctionnalités existantes ou en cours de développement).

Supprimer des warnings

- ▶ Compiler le projet dans votre environnement.
- ▶ Corriger les warnings à la compilation si il y en a.
- ▶ Ouvrir une issue et proposer le patch (selon les outils utilisés par le projet).

Ajouter des commentaires

- ▶ Dans un projet collaboratif et/ou conséquent, les commentaires sont importants.
- ▶ Commenter les parties du code que vous avez eu du mal à comprendre à première lecture.
- ▶ Proposer d'intégrer ces commentaires au code du projet.

- ▶ Souvent des bugs corrigés mais pas fermés.

Corriger un bug

- ▶ Trouver un bug soi-même (souvent difficile ou du moins aléatoire).
- ▶ Choisir un bug déjà rapporté.
- ▶ Signaler qu'on travaille dessus / demander si personne ne le fait déjà.
- ▶ Travailler dessus.
- ▶ Soumettre.

- ▶ Chercher des projets en rapport avec ses compétences, et potentiellement avec des bugs marqués comme “facile”.
 - <https://up-for-grabs.net/>
 - <https://github.com/MunGell/awesome-for-beginners>
 - <https://goodfirstissue.dev/>
 - <https://bugs.launchpad.net/inkscape/+bugs?field.tag=easy-fix>
 - <https://wiki.documentfoundation.org/Development/EasyHacks>
 - <https://github.com/python/cpython/issues?q=is%3Aopen+is%3Aissue+label%3Aeasy>
 - https://bugs.kde.org/buglist.cgi?bug_status=UNCONFIRMED&bug_status=CONFIRMED&query_format=advanced&short_desc=%5EJJ&short_desc_type=regexp
 - <https://github.com/SimpleMobileTools>