

Cours Logiciels Libres

Gouvernance d'un projet open source (suite)

Théo Zimmermann

Université Paris-Cité & Inria

Vendredi 18 mars 2022

Cooptation

- Acquérir des **droits en écriture** (commit) sur le projet est un signe de confiance de la part des autres mainteneurs.
- En général, on ne les accorde qu'à des **contributeurs réguliers**.
- Les mainteneurs (ou le dictateur, ou le PMC) **décident quand proposer** à des contributeurs d'acquérir ces droits. Cela peut faire suite à une discussion dont il est naturel qu'elle ait lieu **en privé**.
- Il peut y avoir une distinction entre des **droits limités** (maintenance d'un composant) et des droits globaux (sur tout le projet).
- Le PMC (ou équivalent) peut être renouvelé également par cooptation ou être élu (par un électorat composé des committers ou plus vaste).

Exemple : projet Coq

- Les créateurs du projet ne sont plus actifs depuis longtemps.
- **Équipe cœur** (actuellement 11 membres), dont un “coordinateur du projet”.
- **Mainteneurs** de composants avec le pouvoir d’approuver / rejeter / intégrer les changements aux **composants qui les concernent** (35 mainteneurs en tout).
- C’est l’équipe cœur qui propose en général à des contributeurs réguliers de devenir des mainteneurs.
- L’équipe cœur est renouvelée par **cooptation**. En général, on commence par être mainteneur de composant avant d’intégrer l’équipe cœur.

Exemple : projet Python

- Guido van Rossum (créateur du langage) était le BDFL jusqu'à ce qu'il **démissionne** en 2018.
- La gouvernance actuelle (inspirée de Django) est composée :
 - D'une **équipe cœur** (96 membres actifs).
 - D'un **comité de direction** (5 membres).
- Les nouveaux membres de l'équipe cœur doivent être validés par un vote de 2/3 des membres actifs.
- Le comité de direction est **élu par l'équipe cœur** et renouvelé régulièrement (~ tous les ans). Il a un fort **pouvoir décisionnaire**, à n'utiliser qu'en dernier recours.
- Pas plus de deux membres du comité de direction ne peuvent avoir le même employeur.

Le consensus paresseux

- Quel que soit le modèle exact de gouvernance d'un projet open source, il est standard que la plupart des décisions soient prises par **consensus paresseux** :
 - Les oppositions doivent être **explicites**.
 - Les silences sont considérés comme des approbations.
 - Les oppositions sont prises au sérieux et quasiment équivalentes à des **vetos** (selon par qui elles sont émises).
- Selon l'importance de la décision : **processus + ou – formel**.
 - Le – formel : agir (pousser un changement) et annuler l'action en cas de protestation (*revert*).
 - Le + formel : déclarer la décision qui va être prise si personne ne s'y oppose pendant une période définie (de plusieurs jours), "*Final Comment Period*".

Les PEP / RFC

Certains projets (dont des langages de programmation comme Python, Go et Rust) ont adopté un modèle de décision pour les **modifications importantes** au code (ou à la gouvernance) reposant sur des **discussions préalables sur des documents**, PEP (Python Enhancement Proposals) ou RFC (Request For Comments) :

Les PEP / RFC

Certains projets (dont des langages de programmation comme Python, Go et Rust) ont adopté un modèle de décision pour les **modifications importantes** au code (ou à la gouvernance) reposant sur des **discussions préalables sur des documents**, PEP (Python Enhancement Proposals) ou RFC (Request For Comments) :

- **Un projet de modification est présenté** avec des motivations, des explications sur les choix effectués, les alternatives et les conséquences qui ont été considérées.

Les PEP / RFC

Certains projets (dont des langages de programmation comme Python, Go et Rust) ont adopté un modèle de décision pour les **modifications importantes** au code (ou à la gouvernance) reposant sur des **discussions préalables sur des documents**, PEP (Python Enhancement Proposals) ou RFC (Request For Comments) :

- **Un projet de modification est présenté** avec des motivations, des explications sur les choix effectués, les alternatives et les conséquences qui ont été considérées.
- Une **discussion** a lieu sur les forums / mailing lists qui peut conduire à la **mise à jour du document**.

Les PEP / RFC

Certains projets (dont des langages de programmation comme Python, Go et Rust) ont adopté un modèle de décision pour les **modifications importantes** au code (ou à la gouvernance) reposant sur des **discussions préalables sur des documents**, PEP (Python Enhancement Proposals) ou RFC (Request For Comments) :

- **Un projet de modification est présenté** avec des motivations, des explications sur les choix effectués, les alternatives et les conséquences qui ont été considérées.
- Une **discussion** a lieu sur les forums / mailing lists qui peut conduire à la **mise à jour du document**.
- Un processus de décision (ex. recherche d'un consensus paresseux, validation par l'équipe cœur, etc.) permet d'**entériner la proposition**, qui pourra ensuite être implémentée par des volontaires.

Documenter les processus de gouvernance

Documenter les processus est important pour :

- aider les nouveaux contributeurs à **comprendre** comment les décisions sont prises ;
- servir de **référence** en cas de doute / de conflit.

Il est commun que le **processus** de renouvellement des committers ne soit pas documenté, mais qu'il soit même très **flou** pour les committers eux-mêmes. Cela peut conduire à un **manque de renouvellement**.