

Pour un GitHub plus démocratique et efficace

GitHub est aujourd'hui la plus dynamique forge de développement de logiciels libres. Mais n'y aurait-il pas, dans sa conception même, quelques problèmes de gouvernance et de circulation du code qui menacent l'efficacité, voire la viabilité, des projets ?

Remarque : Pull request, issue, commit... nous présupposons que vous êtes familier avec le vocable GitHub, mais si un gentil lecteur veut nous les préciser dans les commentaires, qu'il/elle n'hésite surtout pas ☐

De la citoyenneté dans le développement de logiciels open source

On Citizenship in Open-source software development

Christophe Maximin - 8 mai 2013 - Blog personnel

(Traduction : ProgVal, Melchisedech, nano-plink, TheCamel, Al + anonymes)

Comment GitHub peut révolutionner la question en donnant le pouvoir aux utilisateurs dans les dépôts auxquels ils contribuent.

TL;DR : En donnant un véritable statut social aux personnes contribuant à un dépôt, GitHub résoudrait le problème des projets-zombies ayant une communauté éparpillée. En permettant à ces citoyens de collaborer réellement les uns avec les autres, et non avec le seul propriétaire, les dépôts seront vivants tant que leur communauté existera, de manière complètement auto-régulée.

L'année a très bien commencé pour GitHub. Après avoir levé cent millions de dollars d'Andreessen-Horowitz et atteint les trois millions d'utilisateurs en janvier (3,4 millions et plus à présent), ils sont sur une dynamique qu'il sera difficile d'arrêter.

Néanmoins, le service a aussi ses défauts, et si certaines personnes pointent du doigt de tous petits problèmes liés aux services et aux applications, le problème que je m'appête à décrire touche à la nature même de nos interactions sur la

plate-forme.

1. État actuel d'un dépôt



Chaque dépôt que vous créez est un petit pays avec une très faible population : 1 habitant, vous, le créateur/roi/commandant suprême.

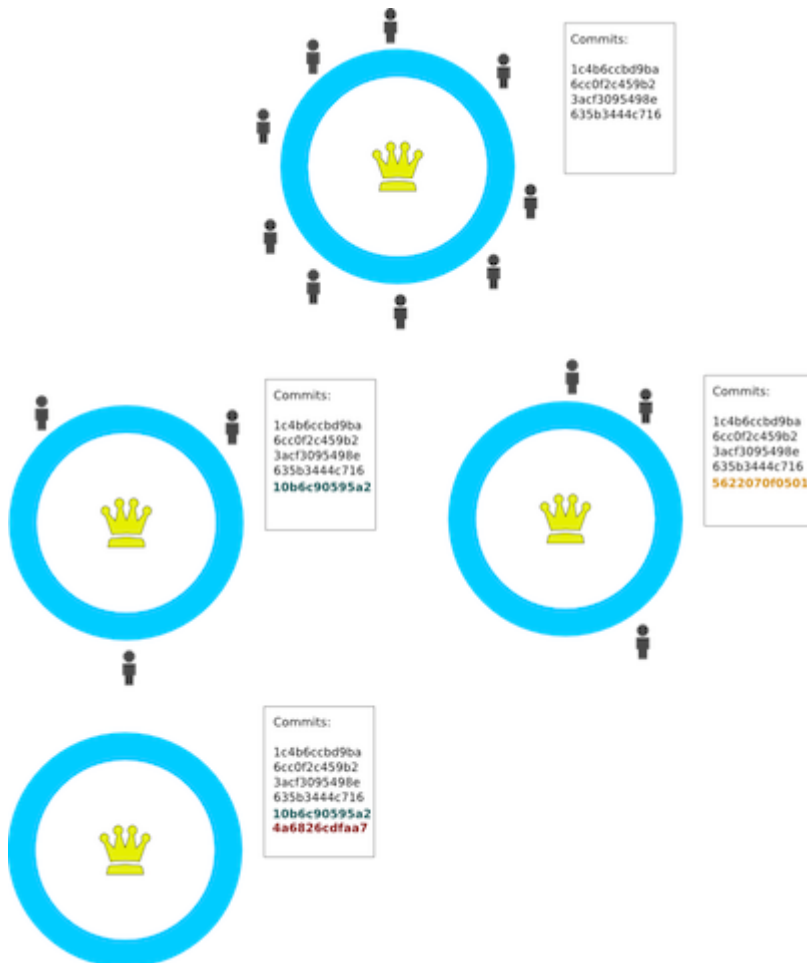
Même si votre dépôt a des centaines de rapports de bugs créés par d'autres, et des centaines de *pull requests*, il n'y a qu'une seule personne aux commandes.

Bien sûr, vous pouvez ajouter des collaborateurs à votre dépôt, mais ils ne seront que des collaborateurs, des membres du cabinet, choisis juste parce que vous le souhaitez. Bien sûr, dans le cas des organisations, vous pouvez ajouter des co-commandeurs suprêmes.

Mais c'est tout. Vous n'atteindrez probablement pas cinquante collaborateurs/membres; même si votre dépôt est vraiment populaire, et même si des centaines de personnes y contribuent. Est-ce que cela vous paraît normal ?

Ce ne serait pas un problème si ce n'était pas la cause de...

2. La fragmentation des dépôts et de leurs fonctionnalités



J'ai vu la chose suivante arriver bien trop souvent :

- Le développeur abandonne graduellement un projet à cause de nouveaux engagements dans sa vie, ou juste parce qu'il n'est plus intéressé. Et donc il regarde les *pull requests* une fois par mois, peut être. Le projet semble vivant. Il ne l'est pas.
- Le développeur est dépassé par les rapports de bugs et les *pull requests* qu'il reçoit. Et bien qu'il sache qu'il a une solide communauté autour de ce projet, il ne peut pas juste ajouter quelqu'un comme collaborateur car il devra quand même lire chaque ligne pour être sûr que tout est en ordre. Et donc, il regarde les *pull requests* une fois par mois, peut être. Le projet semble vivant. Il ne l'est pas vraiment.

Et vous vous dites : « Qui s'en soucie ? N'importe qui peut forker le dépôt et donner une nouvelle vie au projet autre part ! »

Bien entendu, mais combien de fois avez-vous vu cela se faire réellement ?

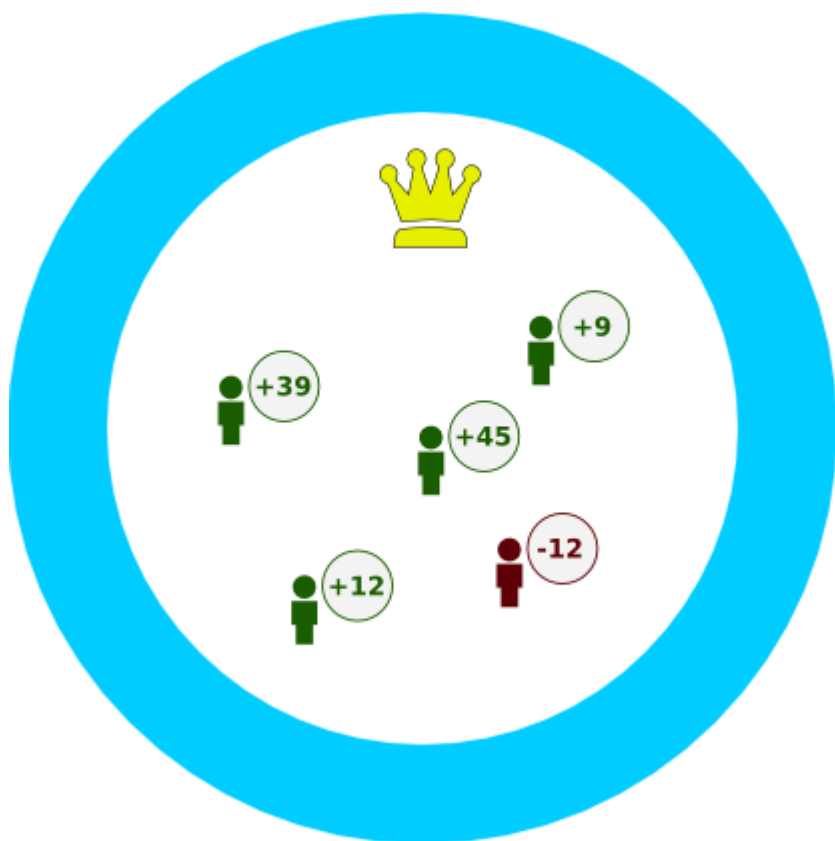
La plupart du temps, les gens forkent le projet pour régler « le » bug qu'ils voulaient régler, et c'est tout.

Maintenant, si vingt personnes agissent de la sorte, cela devient une vraie tragédie : le projet est en fait encore mis à jour, mais à vingt autres endroits. Vous devrez fusionner les *commits* de 20 dépôts différents pour être à jour de toutes les nouvelles choses cools que vous pouvez faire avec le projet. Mais vous ne le ferez jamais. Certains forks sont incompatibles de toute façon.

D'autre part, comme le projet « semble vivant », personne ne se presse pour essayer de le remplacer. La léthargie se poursuit alors encore et encore, et va créer la confusion chez n'importe quel nouveau venu quant à l'état du projet, sur l'emplacement des dernières mises à jour, et sur leur éventuelle acceptation par la communauté.

Je ne vais pas donner de noms (ce n'est pas bien de pointer du doigt publiquement) , mais je suis sûr que vous voyez à quoi je fais référence.

3. Le pouvoir au peuple, le pouvoir à la cité



Sans entrer dans un débat sur les multiples définitions du mot citoyenneté, vous trouverez ici une liste de quelques fonctionnalités qui en feront une réalité. Rien de ce qui est listé ici n'est absolu, et ce sera à l'administrateur de définir les règles.

- Tout le monde peut voter pour une *issue*.
- Tout le monde peut voter pour une *pull request*, avec un *merge* automatique quand une majorité (ou quelque chose d'autre à définir) est atteinte.
- Les citoyens se voient attribuer des « points de karma » suivant les votes positifs ou négatifs qu'ils reçoivent sur leurs *commits* et leurs réponses aux *issues*. Les citoyens ont un total de points pour ce dépôt, et pour le reste de GitHub.
- Tous les *commits* qui sont approuvés par le peuple vont dans un branche spécifique préfixée « *par_le_peuple_** » .
- Les administrateurs ont toujours le droit de veto sur ce qu'ils veulent, et peuvent complètement couper ce mode « auto-pilote » .

Conclusion

Il est temps que les gens qui contribuent à des projets acquièrent enfin une réelle existence. Il n'y a vraiment rien à perdre, et cela semble pour moi être une évolution naturelle et inévitable de toute façon.

La question est : combien de temps devons nous attendre ?