

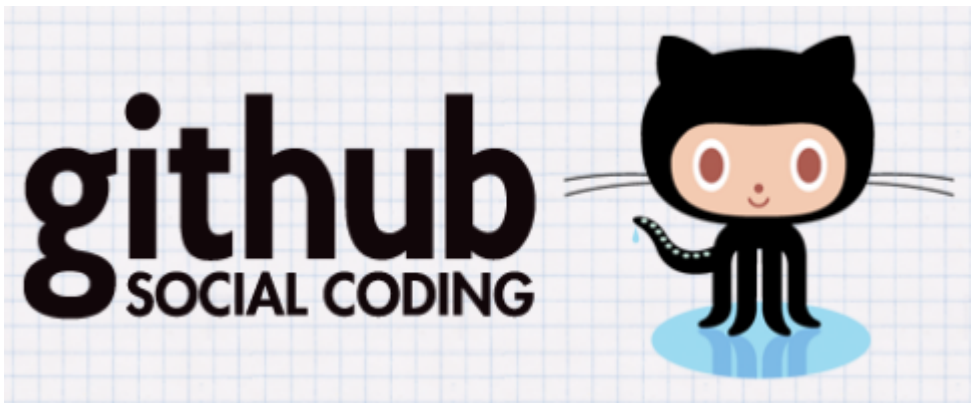
# La génération GitHub

GitHub a beau être une plateforme non libre de projets libres, force est de constater que cette « forge sociale » est devenue en quelques années l'un des centres névralgiques de la communauté.

Avec sa facilité d'usage, son appel permanent au fork et l'individuation des contributions, GitHub a permis à plus de monde de participer tout en ouvrant le Libre au delà du logiciel puisqu'il n'y a pas que du code proprement dit dedans (cf la liste de l'article traduit ci-dessous).

À tel point que certains n'hésitent pas à y voir un modèle pertinent pour toutes sortes de choses à commencer par la... démocratie !

Et si une génération toute entière était effectivement en train de naître sous nos yeux ?



## La génération Github : Pourquoi vous et moi pouvons désormais faire de l'Open Source

**The GitHub Generation: Why We're All in Open Source Now**

*Mikeal Rogers - 7 mars 2013 - Wired Opinion*

*(Traduction : Moosh, Sphinx, Peekmo, Chopin, goofy, misc, Uflex + anonymes)*

GitHub a été conçu pour être une plate-forme de collaboration logicielle ouverte, mais c'est devenu une plate-forme pour déposer beaucoup plus de choses que du simple code. Elle est maintenant utilisée par des artistes, des créateurs, des

propriétaires de maisons et des tas d'autres gens, par des entreprises entières... et même par des municipalités.

« N'importe qui peut maintenant changer les données quand de nouvelles pistes cyclables sont aménagées, quand de nouvelles routes sont construites ou quand de nouveaux immeubles sont construits » a annoncé récemment la ville de Chicago. Les gens planifient leurs projets de rénovation de maison sur GitHub. Un cabinet d'avocats a annoncé il y a quelques jours qu'il postait des documents juridiques pour des start-ups sur GitHub. Quelqu'un a même publié toutes les lois d'Allemagne sur GitHub l'année dernière (avec, s'il vous plaît, déjà 17 *pull requests* pour des modifications).

Bien sûr, GitHub reste majoritairement toujours utilisé par les programmeurs et développeurs qui font voler des AR.Drones avec Node.js ou construisent des sites web avec jQuery. Mais de plus en plus de gens passent de consommateurs à producteurs, et ils redéfinissent ainsi la culture de l'open source. Je crois que GitHub transforme l'open source comme l'internet a transformé l'industrie de la publication : un fossé culturel est en train de se creuser entre l'ancienne génération de gros projets libres et la nouvelle génération d'amateurs de projets libres d'aujourd'hui.

## **La révolution ne sera pas centralisée**

Quand la plupart des gens entendent « open » source, ils pensent démocratie, distribution, égalité : tout le monde construit des choses pour que tout un chacun les utilise.

Mais cela n'a pas toujours été le cas. La plupart des logiciels open source ont été créés et maintenus par une classe privilégiée et protégée, les développeurs professionnels, qui interagissaient avec d'autres développeurs très semblables (ils sont pourtant suffisamment différents pour avoir de belles disputes).

Avant GitHub, je passais beaucoup de temps à penser et à discuter de la meilleure façon de gérer des projets open source parce que la coordination représentait un coût important d'un projet open source. Si important que lorsqu'un projet réussissait et développait une communauté assez grande, il était logique que le projet grandisse plutôt qu'il ne se fracture en projets plus petits. Mais plus le projet du logiciel devenait grand et complexe, plus il était difficile d'y contribuer. Ainsi, un choix de membres, les *committers* - étaient assignés à la gestion et à la

production du projet. Cela menait souvent à des ruptures séparant ceux qui produisaient le projet et ceux qui les utilisaient.

GitHub a comblé ce fossé en faisant de l'open source quelque chose de bien plus décentralisé. C'est devenu davantage centré sur les individus que sur le projet.

La façon d'utiliser GitHub est très personnelle. Une personne (je suis [github.com/mikeal](https://github.com/mikeal)) a un compte, et tout ce qu'elle publie existe à un niveau en dessous d'elle. Si quelqu'un veut corriger quelque chose, il suffit de « forker » le projet, ce qui place une copie sous son propre compte.

Cette façon de travailler est très stimulante : elle encourage les individus à corriger les problèmes et à prendre possession des correctifs au même niveau que le projet de départ. Cela donne également à chacun une identité dans cette nouvelle culture du libre. GitHub est actuellement le premier fournisseur d'identité pour la production collaborative sur internet pour faire plus que du développement de code.

J'ai contribué à des projets libres depuis plus de 10 ans, mais ce qui est différent maintenant est que je ne suis pas un membre d'un de ces projets, je suis un simple utilisateur, et contribuer un peu est devenu une petite partie du rôle d'un utilisateur. Des petites interactions entre moi et les mainteneurs de projets arrivent plusieurs fois par semaine sur tout type de projet que j'utilise. Et ça arrive encore plus souvent dans l'autre sens : des gens dont je n'ai jamais entendu parler m'envoient des petits bouts de code sur les petits projets que j'ai publiés.

## **La décentralisation comme démocratie**

Les premières versions de GitHub ont très bien fait une chose : rendre la publication de votre code beaucoup plus facile (que la non-publication). Ceci était suffisant pour que beaucoup de projets connus, notamment Ruby on Rails, migrent sur GitHub presque immédiatement.

Mais ce qui s'est passé après est encore plus intéressant : les gens ont commencé à tout publier sur GitHub. Pousser du code est presque devenu une habitude, comme tweeter. En abaissant la barrière pour entrer et rendant plus facile la contribution à l'open source, GitHub a élargi la production collaborative aux utilisateurs occasionnels.

Aujourd'hui un vaste choix de logiciels simples et compréhensibles est accessible à une catégorie de gens créatifs qui n'avaient jusqu'alors pas les compétences techniques requises pour participer à des projets open source par le passé.

Ce mélange des relations entre les producteurs, les contributeurs et les consommateurs valorise naturellement les projets plus petits et plus faciles à comprendre — et a conduit à de nombreuses contributions. Au cours du mois de septembre 2012 par exemple, la moitié des utilisateurs actifs de GitHub qui ont poussé au moins un *changeset*, l'ont fait moins de cinq fois, avec 22% (environ 44 000 personnes) qui ont poussé seulement un seul *changeset* ce mois-ci.

L'accès de l'open source aux amateurs présente certains avantages évidents.

## **Faciliter les usages**

Un des problèmes récurrents, avec le logiciel open source, a été la qualité des finitions. La documentation, le design des sites web et l'ergonomie en général ont toujours été un problème — spécialement par rapport à de nombreux concurrents propriétaires.

Mais maintenant, avec les facilités de collaboration, des utilisateurs moins portés sur la technologie et la connaissance du code peuvent plus facilement participer à améliorer les logiciels sur lesquels ils travaillent (ce qui peut être des petites choses comme l'humanisation des messages d'erreur de codage ou de légers changements graphiques en une ligne de CSS qui optimisent le rendu des sites web des navigateurs, anciennes versions incluses, et sur les téléphones mobiles).

Dans le nouvel open source, les gens veulent utiliser la technologie sans avoir besoin de devenir des experts. La facilité d'utilisation est plus valorisée que jamais.

## **Éviter de réinventer la roue**

Les développeurs aiment les défis et plus ils ont de chances de les relever, plus leurs solutions peuvent être astucieuses. C'était parfait lorsque les utilisateurs de ces solutions étaient eux aussi des gens très compétents techniquement comme ceux qui prenaient plaisir à résoudre astucieusement ces anciens problèmes.

Mais les amateurs préfèrent les solutions qu'ils peuvent tenir pour acquises : une fois qu'un problème est résolu, ils reviennent rarement en arrière pour le

réexaminer. Et dans la mesure où les amateurs ne créeront qu'à partir des solutions les plus compréhensibles, cela contraint les développeurs à élaborer des solutions simples qui rendent les problèmes complexes plus faciles à appréhender.

## **Soutenir un écosystème plus vaste**

Node.js, projet dans lequel je suis activement impliqué, définit des modèles suffisamment simples pour que les gens puissent écrire de petites bibliothèques indépendantes et les publier à leur gré. Tous ceux qui s'impliquent dans l'écosystème peuvent en tirer profit sans coordination. C'est le pôle inverse de l'énorme pile verticale qui accompagne des tas d'outils et fonctionnalités (tels que dans les systèmes intégrant des plugins, comme Ember, Dojo et YUI) qui sont nécessaires pour réussir à développer dans des environnement propriétaires (pensez à Cocoa et au développement pour iOS). Dans les environnements ouverts, tels que Node.js sur GitHub, nous constatons que des API bien plus légères peuvent facilement tirer parti du reste de l'écosystème sans coordination. Moins il y a de coordination entre les développeurs et les bibliothèques et plus nous pouvons créer de la valeur.

GitHub a donné les capacités à une nouvelle génération de collaborer, de créer, de produire. Beaucoup de développeurs regretteront l'abandon des normes culturelles précédentes, telles que le statut des *committers* (ceux qui sont autorisés à envoyer le code sur le dépôt) ou la bonne vieille guerre pour le choix de la bonne licence — mais l'avenir est déjà entre les mains d'une nouvelle génération qui a évolué.

Ce n'est pas un simple outil : c'est à la naissance d'une nouvelle culture à laquelle nous assistons.