

# RFC 8874 : Working Group GitHub Usage Guidance

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 août 2020

Date de publication du RFC : Août 2020

<https://www.bortzmeyer.org/8874.html>

---

De nombreux groupes de travail de l'IETF utilisent GitHub pour coordonner leur travail dans cet organisme de normalisation. Ce nouveau RFC donne des règles et des conseils à suivre dans ce cas. Personnellement, je déplore qu'il ne contienne pas de mise en garde contre l'utilisation d'un service centralisé extérieur à l'IETF.

L'IETF développe des normes pour l'Internet, notamment la plupart des RFC. Ce travail met en jeu des documents, les brouillons de futurs RFC, des discussions, des problèmes à résoudre. Les discussions sont assez complexes, durent souvent des mois, et nécessitent de suivre pas mal de problèmes, avec beaucoup d'éléments à garder en tête. Bref, des outils informatiques peuvent aider. Mais attention, l'IETF a des particularités (cf. RFC 2418<sup>1</sup>). D'abord, tout doit être public, le but étant que le développement des normes soit aussi transparent que possible. Et il ne s'agit pas que de regarder, il faut que tous les participants (l'IETF est ouverte) puissent intervenir. Ensuite, tout doit être archivé, car il faudra pouvoir rendre compte au public, même des années après, et montrer comment l'IETF est parvenue à une décision. (D'autres SDO, comme l'AFNOR, sont bien plus fermées, et n'ont pas ces exigences d'ouverture et de traçabilité.)

Traditionnellement, l'IETF n'avait que peu d'outils de travail en groupe et chacun se débrouillait de son côté pour suivre le travail tant bien que mal. Mais les choses ont changé et l'IETF a désormais une importante série d'outils, aussi bien les « officiels », maintenus par un prestataire extérieur, AMSL <<https://www.amsl.com/>>, que des officieux, disponibles sur et maintenus par des volontaires. Les outils officiels de l'IETF, mis à la disposition de tous, comportent notamment :

- Le service d'accès et de recherche dans les archives des listes de diffusion, MailArchive <<https://mailarchive.ietf.org/>>.

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2418.txt>

- 
- Le service de suivi des documents DataTracker <<https://datatracker.ietf.org/>> (regardez par exemple où en est HTTP/3 <<https://datatracker.ietf.org/doc/draft-ietf-quic-http/>>).

Il n'y a pas par contre de VCS (dans les outils officiels, il y a un service Subversion) ou de mécanisme de suivi de tickets (il y a un Trac officiels en , il est très peu utilisé).

Un VCS est pourtant un outil très utile, et pas seulement pour les programmeurs. Travaillant en général sur des fichiers texte quelconques, il permet de coordonner le travail de plusieurs personnes sur un document, que celui-ci soit du code source ou bien un "*Internet-Draft*". C'est le cas par exemple du VCS décentralisé git, très utilisé aujourd'hui.

On peut utiliser git tout seul, d'autant plus que le fait qu'il doit être décentralisé ne nécessite pas d'autorité de coordination (c'est ainsi qu'est développé le noyau Linux, par exemple). Mais beaucoup de programmeurs (ou d'auteurs d'"*Internet Drafts*") utilisent une forge complète, intégrant un VCS mais aussi un système de gestion de rapports, un wiki et d'autres fonctions. La plus connue de ces forges est GitHub, propriété de Microsoft. Comme souvent avec les réseaux sociaux (GitHub est le Facebook du "*geek*"), la taille compte : plus il y a de gens sur le réseau social, plus il est utile. C'est pour cela qu'il y a des fortes pressions à la centralisation, et une difficulté, en pratique, à partir, sauf à perdre les possibilités de coopération (c'est pour cela que prétendre qu'il y a consentement des utilisateurs aux pratiques déplorables des réseaux sociaux centralisés est une farce : on n'a pas le choix). L'utilisation de GitHub n'est évidemment pas obligatoire à l'IETF mais elle est répandue. Regardez par exemple le groupe de travail QUIC <<https://github.com/quicwg/>>, celui sur HTTP <<https://github.com/httpwg/>>, ou celui sur les portails captifs <<https://github.com/capport-wg/>>. GitHub est également utilisé en dehors des groupes de travail, par exemple pour les résultats et présentations des hackathons <<https://github.com/IETF-Hackathon/>>. Le but de ce RFC est de fournir quelques règles utiles (mais facultatives) pour les groupes qui travailleront sur GitHub. Un autre RFC, le RFC 8875, est plus concret, donnant des règles précises de configuration de GitHub pour un groupe de travail.

L'introduction du RFC rappelle juste en passant qu'il existe d'autres forges comme Bitbucket et du logiciel libre qui permet d'installer diverses forges locales. Ainsi, le logiciel de GitLab permet d'installer des forges indépendantes comme Framagit <<https://framagit.org/>>. (J'ai récemment migré tous mes dépôts vers un GitLab <<https://www.bortzmeyer.org/github-to-gitlab.html>>.) C'est Framagit qui est utilisé pour le développement de la norme sur la QNAME minimisation <<https://framagit.org/bortzmeyer/rfc7816-bis>>, le RFC 9156, qui a remplacé le RFC 7816. Mais GitHub est de loin le plus utilisé et, par exemple, pour le successeur du RFC 7816, quasiment tous les contributeurs avaient été obligés de se créer un compte sur Framagit, ils n'en avaient pas avant. Le RFC ne rappelle même pas les dangers qu'il y a à utiliser un service centralisé, géré par une entreprise privée. Si GitLab est mentionné dans le RFC, la possibilité d'une instance IETF (gérée par l'actuel prestataire qui administre les ressources IETF, AMSL <<https://www.amsl.com/>>) n'apparaît pas. Simple question d'argent ou problème plus fondamental ?

Bon, au boulot, maintenant, quelles sont les règles listées par ce RFC ? (Rappel : ce RFC reste à un haut niveau, les instructions précises sont dans le RFC 8875.) D'abord, comment décider d'utiliser (ou non) GitHub (section 3 du RFC) ? Fondamentalement, ce sont les présidents du groupe qui décident, après consultation avec les directeurs de la zone (une zone regroupe plusieurs groupes de travail). Ils vont également définir les conditions d'utilisation (par exemple, décider d'utiliser le système de rapports - "*issues*" - ou pas). Notez que, même si le groupe en tant que tel n'utilise pas GitHub, certains contributeurs peuvent évidemment s'en servir pour leurs besoins propres. D'une manière générale, le travail fait sur GitHub n'a pas d'autorité particulière, c'est juste un élément parmi d'autres pour le groupe. Ainsi, ce n'est pas parce qu'un ticket est fermé sur GitHub que le groupe de travail est lié par cette fermeture et ne peut plus discuter le point en question.

Le groupe peut ensuite créer un dépôt par document, ce qui est recommandé (c'est ce que fait le groupe capport <<https://github.com/capport-wg>>) ou bien, s'il préfère, un seul dépôt pour tous les documents du groupe (sans aller jusque là, le groupe QUIC met tous les documents de base du protocole <<https://github.com/quicwg/base-drafts>> dans le même dépôt).

Les documents sont a priori en XML, format standard des documents IETF (cf. RFC 7991) mais on a le droit d'utiliser Markdown (cf. RFC 7328). Par contre, les formats doivent être du texte, pas de binaire comme avec LibreOffice, binaire qui passerait mal avec git (pas moyen de voir les différences d'une version à l'autre).

Ensuite, un peu de paperasserie (section 2 du RFC), avec les règles administratives. Le RFC recommande de créer une organisation dans GitHub pour chaque groupe de travail qui utilise GitHub (il n'y a pas d'organisation « IETF »). Les propriétaires de cette organisation doivent être les directeurs de la zone dont fait partie le groupe, et les présidents ou présidentes du groupe. Les auteurs des documents doivent évidemment avoir un accès en écriture. Les dépôts du groupe de travail sur GitHub doivent être clairement documentés, indiquant la charte du groupe, sa politique, sa gestion des contributeurs (fichier CONTRIBUTING, que GitHub met en avant), etc. Et il faut évidemment pointer sur la politique générale de l'IETF, le "Note Well" <<https://www.ietf.org/about/note-well/>> (qui est tout le temps cité dans les réunions physiques, mais qui doit également être connu des gens qui n'interagissent que via GitHub). En sens inverse, le dépôt GitHub doit être clairement indiqué sur les pages du groupe sur le site de l'IETF, et sur les "Internet-Drafts" produits.

Un des intérêts d'une forge comme GitHub est qu'on dispose de plusieurs moyens de contribuer à un projet. Lesquels utiliser pour des documents IETF (section 4 du RFC)? Il y a d'abord le système de suivi des questions. Il permet de noter les questions en cours et les décisions prises peuvent être inscrites dans le ticket, avant de le fermer. D'un coup d'œil, on peut voir facilement le travail en cours, et ce qui reste à faire. Dans chaque ticket, on peut voir l'ensemble des éléments liés à une question. Le système de suivi de questions ("issue tracker") de GitHub permet bien d'autres choses, comme l'affectation d'un ticket à une personne. Un service très utile est l'étiquetage des questions (« rédaction » pour un texte mal écrit, « technique » pour un problème technique, etc). Comme dans la plupart des cas, le RFC n'impose pas une politique de gestion des étiquettes, seulement le fait qu'il y ait une politique et qu'elle soit communiquée aux participants. Même chose pour la politique de fermeture des tickets (un sujet parfois sensible).

Une fonction qui a beaucoup contribué à la popularité du GitHub est la possibilité de "pull request" (qui a une bonne traduction en français? Alexis La Goutte suggère « demande d'intégration »), c'est-à-dire d'enregistrer une série de modifications qui sont collectivement soumises à l'approbation ou au rejet d'un responsable. Cela permet de proposer une contribution sans avoir l'autorisation d'écriture dans le dépôt, tout en permettant une grande traçabilité des contributions. C'est la méthode conseillée par le RFC pour soumettre des contributions significatives, notamment en raison de cette traçabilité. Par contre, il n'est pas conseillé de discuter de questions complexes dans les commentaires de la "pull request"; comme la requête peut être modifiée, les commentaires risquent de devenir décalés par rapport au dernier état de la "pull request".

GitHub permet d'avoir un flux de syndication pour les dépôts. C'est ce que j'utilise personnellement pour suivre les activités des dépôts qui m'intéressent. Mais il existe d'autres méthodes comme l'outil github-notify-ml <<https://github.com/dontcallmedom/github-notify-ml>>, très utilisé à l'IETF.

J'ai dit plusieurs fois que ce RFC n'imposait pas **une** façon d'utiliser GitHub pour un groupe de travail IETF. Il y a plusieurs politiques possibles, chaque groupe de travail peut faire différemment, l'important étant que les participants au groupe soient au courant. Pour faciliter le choix, la section 5 du RFC propose un jeu de politiques typiques, parmi lesquelles on peut choisir. Elles sont classées de la plus élémentaire, à celle qui utilise le plus les possibilités de GitHub :

- Dans le cas le plus simple, le groupe travaille comme avant, il ne se sert que GitHub que pour stocker les documents, notamment les *"Internet-Drafts"* et leur historique.
- Plus élaborée, une politique avec utilisation du système de suivi des questions, mais où les discussions ont toujours lieu ailleurs que sur GitHub, les tickets ne stockant que les questions en cours et leur résolution. (Comme le note le RFC, c'est difficile à garantir, les discussions s'invitent souvent sur les tickets.)
- Encore plus githubienne, la politique où cette fois la discussion a officiellement lieu sur les tickets GitHub. La liste de diffusion (le principal outil officiel des groupes de travail IETF) a le dernier mot mais le gros du travail a été fait sur le système de gestion de tickets. Attention : cela veut dire qu'une bonne partie des explications et justifications à un RFC vont rester sur une plate-forme privée, qui peut changer sa politique à sa guise, ce qui pourrait faire perdre à l'IETF une partie de son histoire (la section 10 du RFC revient sur cette question). Le RFC ne contient pas de mise en garde à ce sujet, alors que le problème s'est déjà produit <<https://www.bortzmeyer.org/ietf-trust.html>>.

La section 5 discute également des politiques de nommage pour les étiquettes. Au minimum, il faut des étiquettes pour différencier les détails (« coquille dans le deuxième paragraphe ») des questions de fond. Ensuite, les étiquettes peuvent être utilisées pour :

- L'état d'un ticket (« en discussion », « consensus approximatif », « repoussé à une future version du protocole »).
- La partie du protocole concernée.
- Ou d'autres cas comme par exemple le marquage des tickets considérés comme complètement hors-sujet.

Une notion importante à l'IETF est celle de consensus. L'IETF ne vote pas (dans une organisation sans adhésion explicite, qui aurait le droit de vote ?) et prend normalement ses décisions par consensus (cf. RFC 2418). Pour estimer s'il y a consensus ou pas, les présidents du groupe de travail peuvent utiliser tous les moyens à leur disposition (section 7 du RFC) dont GitHub, mais doivent s'assurer, sur la liste de diffusion officielle du groupe, qu'il s'agit bien d'un consensus. (Certains participants ou participantes peuvent ne pas utiliser GitHub, et il faut quand même tenir compte de leur avis.)

Le RFC note aussi (section 8) que, parmi les fonctions de GitHub, l'intégration continue peut être très utile. Par exemple, si les documents rédigés comportent des extraits dans un langage formel (comme YANG ou Relax NG), mettre en place un système d'intégration continue permet de détecter les erreurs rapidement, et de présenter automatiquement la dernière version, automatiquement vérifiée, des documents.

La section 9 du RFC donne quelques conseils aux auteurs (*"editors"* dans le RFC car le concept d'auteur est délicat, pour des documents collectifs). Avec GitHub, il y aura sans doute davantage de contributions extérieures (par exemple sous forme de *"pull requests"*), puisque le processus de soumission est plus facile pour les gens qui ne sont pas à l'IETF depuis dix ans. C'est bien l'un des buts de l'utilisation d'une forge publique. Mais cela entraîne aussi des difficultés. Des tickets vont être dupliqués, des *"pull requests"* être plus ou moins à côté de la plaque, etc. Les auteurs doivent donc se préparer à un travail de tri important.

Enfin, la section 10, consacrée à la sécurité, détaille plusieurs des problèmes que pose l'utilisation de GitHub. D'abord, il y a la dépendance vis-à-vis d'une plate-forme extérieure à l'IETF ; si GitHub est en panne, perturbant le fonctionnement d'un ou plusieurs groupes de travail, l'IETF ne peut rien faire. (Le caractère décentralisé de git limite un peu l'ampleur du problème.) Ensuite, il y a la nécessité de faire des sauvegardes, y compris des tickets (qui ne sont pas, eux, gérés de manière décentralisée). Le RFC 8875 donne des consignes plus précises à ce sujet. Et il y a le risque de modifications non souhaitées, si un participant ayant droit d'écriture se fait pirater son compte. Le fait que tout soit public à l'IETF, et que git permette assez facilement de revenir en arrière sur des modifications, limitent l'importance de cette menace mais, quand même, il est recommandé que les auteurs sécurisent leur compte (par exemple en activant la MFA). En revanche, un autre risque n'est pas mentionné dans le RFC, celui du blocage par GitHub des méchants pays. GitHub bloque ou limite l'accès <<https://www.zdnet.com/article/>

github-starts-blocking-developers-in-countries-facing-us-trade-sanctions/> à certains pays et ne s'en cache d'ailleurs pas <<https://help.github.com/en/github/site-policy/github-and-trade-controls>>.

Dernier problème avec GitHub, non mentionné dans le RFC (merci à Alexis La Goutte pour l'observation) : l'IETF travaille de nombreuses années à développer et promouvoir IPv6 et GitHub n'a toujours que les adresses de la version du siècle précédent <<https://dns.bortzmeyer.org/github.com/ADDR>>.