

Le projet Rust et sa gestion collaborative

La gestion d'une communauté de développeurs d'un projet *open source* est assez délicate. Le langage de programmation Rust, conçu et développé de façon ouverte au sein de Mozilla depuis une bonne dizaine d'années, fait largement appel à sa communauté dans sa structure et son mode de développement.

Son originalité par rapport au *management* de projet en entreprise apparaît de façon intéressante dans le témoignage et les réflexions de Mara Bos, une jeune développeuse impliquée dans le projet dont Framalang a traduit ci-dessous les propos.

Article original sur le blog de l'autrice : [Rust is not a company](#)

Traduction Framalang : Fabrice, Côte, goofy

Rust n'est pas une entreprise

par Mara Bos



Mara Bos alias m-ou-se
(son dépôt de code)

L'année dernière, en tant que contributrice occasionnelle au projet Rust, je n'ai pas trop réfléchi à la structure organisationnelle de l'équipe derrière le projet. J'ai vu une hiérarchie d'équipes et de groupes, de chefs d'équipe, etc. Tout comme n'importe quelle autre organisation. Cette année, après m'être davantage impliquée, et devenue membre de l'équipe *Library*¹ puis cheffe d'équipe, j'ai pris le temps de me demander pourquoi nous avons cette structure et à quoi servent ces équipes.

Pas une entreprise

Dans la plupart des entreprises, on trouve des directeurs et des actionnaires et autres trucs dans le genre en haut de la hiérarchie, qui définissent les buts de l'entreprise. Objectifs, jalons, dates limites, et autres choses qui vont sûrement mener l'entreprise vers le but final quel qu'il soit ; généralement l'argent.

Ensuite il existe plusieurs niveaux de *management* répartis en départements et en équipes pour diviser la charge de travail. Chaque niveau prend en charge une part des objectifs et s'assure que sa part est faite, en assignant des tâches aux employés en bas de la hiérarchie, tous travaillant d'une manière ou d'une autre à atteindre l'objectif principal commun.

Alors que la structure derrière un gros projet open source comme Rust peut sembler similaire, vue de loin, c'est souvent complètement l'inverse. Dans un tel projet, les objectifs et buts ne sont pas ceux des équipes d'en haut, mais effectivement ceux des contributeurs.

En tant qu'équipe *library*, nous pourrions par exemple essayer de décider que le mécanisme de formatage (`std::fmt`) devrait être réécrit pour être plus petit et plus efficace. Mais prendre cette décision ne provoque pas la réécriture. Et nous n'avons pas d'employés à qui assigner les tâches. Ce n'est pas

comme ça que ça fonctionne.

Au lieu de cela, un contributeur passionné d'algorithmes de formatage pourrait se manifester, et commencer à travailler sur le problème. Notre travail en tant qu'équipe *library* est de faire en sorte que cette personne puisse travailler. S'assurer que son projet est en accord avec le reste de la bibliothèque standard, relire son travail et fournir un retour utile et constructif. Si davantage de personnes interviennent pour collaborer sur ce projet, mettre en place un groupe de travail pour aider à tout organiser, etc.

Les entreprises ne font pas travailler le premier venu sur quelque chose. C'est ce qui fait que l'*open source* est particulier et si génial, si on s'y prend bien.

Objectifs personnels

Idéalement, l'objectif d'un projet *open source* comme Rust est simplement la combinaison des buts personnels de toutes les personnes travaillant dessus. Et là est la difficulté. Parce que quand une nouvelle personne arrive, on ne lui assigne pas une tâche qui correspond à nos objectifs. Cette personne arrive plutôt avec ses propres objectifs et ses propres idées, les ajoutant à un ensemble déjà assez varié d'objectifs potentiellement conflictuels.

Et c'est pourquoi un projet *open source* piloté par des bénévoles a besoin d'une structure de *management*. On ne peut pas juste mettre ensemble une centaine de personnes avec chacune ses propres buts et espérer que tout se passe bien.

Donc ce que fait le *management*, c'est prendre en considération tous les buts personnels de toutes les personnes travaillant sur un sujet, et essayer de les guider de manière à ce que les choses fonctionnent. Ça peut impliquer le fait de dire non à des idées quand elles seraient incompatibles avec d'autres idées, ou ça peut impliquer beaucoup de discussions pour

harmoniser les idées afin qu'elles soient compatibles. C'est exactement l'opposé de la manière dont fonctionne une entreprise typique, où les objectifs viennent d'en haut, et où le *management* décide de la manière de les répartir et les assigner aux personnes qui effectuent le travail technique.

Alors que beaucoup de projets *open source*, dont Rust, ont un cap ou un plan d'action, ces derniers doivent reposer sur les objectifs des contributeurs individuels du projet pour que ça fonctionne. Dire « notre objectif principal en 2021 est d'améliorer les mécanismes de formatage dans la bibliothèque standard » devrait faciliter le travail des personnes travaillant déjà dessus, et attirer les personnes qui voulaient déjà travailler sur quelque chose de ce genre. Ça devrait les aider parce que nous priorisons toutes les décisions de *management* et les révisions de code dont ils ont besoin. Ça devrait permettre aux personnes de se concentrer et d'avancer davantage. Mais sans ces contributeurs, mettre en place de tels objectifs n'a pas de sens. Contrairement à une entreprise, nous n'avons pas à choisir ce sur quoi les personnes passent leur temps, et nous n'employons pas de personnes auxquelles assigner des tâches.

Et c'est une bonne chose.



Le logo du
langage de
programmation
Rust

C'est la raison pour laquelle les gens veulent travailler sur Rust.

Je ne prétends pas qu'un langage de programmation ne pourrait pas être géré « d'en haut » comme une entreprise le ferait. Beaucoup de langages de programmation ont été et sont développés de cette manière avec beaucoup d'efficacité. Ce que je dis, cependant, c'est que personnellement je ne veux pas que le projet Rust fonctionne de cette manière.

Je ne veux pas **gérer** le département *library* de l'entreprise Rust. Je veux **aider** les personnes qui veulent améliorer les bibliothèques du langage Rust.

Un espace pour s'épanouir

Différents contributeurs ont des objectifs très différents, travaillent avec des méthodes très variées, et ont des besoins très différents des structures de management.

Pour certains d'entre eux, nous avons des processus en place destinés à leur rendre le travail plus facile. Une personne qui veut travailler sur une nouvelle fonctionnalité du langage peut soumettre ses idées via une RFC ² et prendre part dans une discussion avec l'équipe *library* pour des conseils et de l'aide. Quelqu'un qui veut améliorer une grande partie du code du compilateur ³ peut soumettre une proposition de changement majeur (MCP) et en discuter avec l'équipe du compilateur. Et quelqu'un qui veut résoudre un problème dans la bibliothèque standard peut soumettre une proposition de modification et la voir relue et évaluée par des personnes qui connaissent le contexte.

En d'autres termes, nous avons créé un espace pour ces types de contributeurs. De l'espace pour faire leur travail, de l'espace pour obtenir des retours, de l'espace pour obtenir de l'aide, de l'espace pour obtenir une reconnaissance, tout l'espace dont ils ont besoin pour réussir.

Cependant, il existe malheureusement beaucoup de types de contributeurs pour lesquels nous n'avons pas créé un espace,

ou pas le bon espace.

Jusqu'à une époque récente, l'équipe *library* était principalement concentrée sur la conception de l'API. Les problèmes critiques d'implémentation étaient gérés par l'équipe du compilateur par nécessité. Les modifications de code plus modestes étaient relus et évalués par des relecteurs individuels de l'équipe *library*. Mais il n'y avait pas de gestion prévue pour de plus gros changements de code. Ça signifiait qu'il n'y avait pas d'espace pour une personne qui aurait voulu remettre à neuf le code de `std::fmt`. Il n'y avait pas d'équipe qu'elle aurait plus rejointe pour travailler là-dessus, rendant beaucoup plus difficile, voire impossible, d'atteindre son but.

C'est pourquoi j'ai lancé la nouvelle équipe *Library*.

Faire de la place pour quelque chose peut souvent amener à (accidentellement) retirer de la place pour autre chose. Une personne qui n'est pas très impliquée dans le code mais qui veut appliquer son expérience dans la conception d'API et qui se soucie beaucoup de l'interface de la bibliothèque standard ne s'épanouirait pas dans une équipe qui aime avoir des réunions hebdomadaires à propos des problèmes de correction du code et de la manière avec laquelle résoudre ces problèmes avant la prochaine version publique.

Voilà pourquoi nous avons à présent l'équipe *Library API*

Nous avons maintenant aussi une équipe de « Contributeurs à *Library* ». Un endroit pour les personnes qui sont plus impliquées dans le projet que les contributeurs occasionnels, mais qui ne font pas partie des équipes qui prennent les décisions finales. Ça fait de l'espace pour les personnes qui veulent, par exemple, travailler sur seulement un aspect particulier de la bibliothèque, ou aider à relire les modifications proposées. Jusqu'à il y a environ un an, il n'y avait pas d'espace particulier pour qu'une personne relise les

changements ou qu'elle s'implique davantage d'autres manières, sans faire directement partie d'une équipe avec beaucoup plus de responsabilités.

J'ai réussi à effectuer ces changements de structure d'équipe avec l'aide des membres des équipes *Core* et *Library*, qui m'ont mise en capacité de le faire. En retour, ces changements vont, je l'espère, permettre à des membres de ces nouvelles équipes de s'épanouir, et ainsi permettre à encore plus de personnes de contribuer, pour que finalement cela soit bénéfique pour tous les utilisateurs et utilisatrices du langage.



La mascotte de Rust

Continuer à changer

Je n'ai pas l'impression que nous ayons maintenant un espace pour toute personne souhaitant contribuer. Mais une fois que la poussière de la réorganisation se sera dispersée, je pense que le résultat sera une amélioration par rapport à ce que nous avions auparavant, et que ça correspond mieux au projet tel qu'il est aujourd'hui.

« Aujourd'hui » est le mot important ici. Ces équipes sont faites autour des membres actuels et des personnes qui, je pense, pourraient devenir des membres dans un futur proche. Mais ce groupe de personnes et leurs besoins changent, parfois à une vitesse surprenante. Et dans un projet qui est entièrement défini par les personnes qui y contribuent, ça change le projet en lui-même et la structure dont il a besoin,

tout aussi rapidement.

En 2018, l'équipe Library était très impliquée dans l'aide aux auteurs des bibliothèques ⁴ populaires de l'écosystème. L'équipe a publié un ensemble de guides, et va réviser les bibliothèques et travailler conjointement avec leurs auteurs pour les implémenter. Tout cela pour améliorer la cohérence et la qualité de l'écosystème de librairies Rust.

Il y a quelques jours encore, c'était toujours techniquement une partie des buts affichés de l'équipe, même si en pratique ça n'était plus vraiment le cas ; particulièrement depuis qu'Ashley Mannix a quitté le projet il y a un moment. Sans les personnes qui ont pour but personnel de faire que des choses se produisent, les choses ne se produisent pas.

Et c'est très bien comme ça.

Tout le monde a une longue liste de vœux pour Rust, de choses qui ne se font pas parce que personne ne travaille dessus. Nous ne sommes pas une entreprise avec des dates limites et des jalons qu'il nous faut absolument atteindre. Nous sommes un groupe, divers et fluctuant, de personnes qui essaient de gérer tout ça avec passion, en s'efforçant de faire en sorte que tous les efforts aboutissent harmonieusement.

Il y a beaucoup de choses pour lesquelles nous devrions avoir de l'espace, mais pour lesquelles nous n'avons encore de place. Mais si nous continuons à essayer, si nous continuons à effectuer de petites améliorations. À nous adapter. À prendre en compte les personnes autour de nous qui veulent contribuer elles aussi ; si nous continuons à réfléchir à la façon dont nous pouvons les aider, eux et tous les autres. Alors chaque pas sera un pas dans la bonne direction, ainsi Rust prospérera et toutes les nombreuses personnes qui travaillent sur Rust et avec Rust s'épanouiront.



Photo « Rusty but pretty » par Jeanne à vélo, licence CC-BY-SA