Un logiciel libre peut-il se passer d'un dictateur bienveillant à vie ?

Le développement d'un logiciel libre est paradoxal. En théorie, et c'est même ce qui le fonde, il est ouvert à tout le monde mais en pratique son efficacité tire très souvent profit d'une structure qui voit une personne, presque toujours le créateur initial, se dégager de la multitude pour exercer, ce que l'on qualifie faut de mieux, une « dictature bienveillante ».

La « dictature »[1] c'est le fait de choisir seul la ligne directrice du projet en assumant les décisions (prises en concertation mais généralement sans réel processus démocratique). La « bienveillance », c'est la capacité du « dictateur » à écouter et l'attitude par laquelle il manifeste la considération qu'il éprouve envers les contributeurs.

C'est le sujet de la traduction du jour qui penche très clairement en faveur de cette étrange méritocratie qui va à l'encontre d'une organisation plus collective que l'auteur nomme ici « comité ». Avec, quoiqu'il arrive, une question en suspens : et si le « dictateur à vie » venait à ne plus pouvoir travailler sur son projet, ce dernier saurait-il véritablement lui survivre ?

NB : Il est à noter que si l'on accepte de voir le réseau Framasoft dans son ensemble comme une sorte de « gros logiciel libre en développement » alors nous n'échappons pas au débat. Le rôle de dictateur bienveillant, si dictature il y a, étant alors joué conjointement par Pierre-Yves Gosset et moi-même.

Les dictateurs dans les logiciels libres et open source

Dictators in free and open source software

Tony Mobily – 22 juillet 2008 – Free Software Magazine

(Traduction Framalang : Olivier et Penguin)

Certaines personnes remettent en cause l'idée que la plupart (voire la totalité) des projets de logiciels libres ne peuvent pas se passer d'un dictateur bienveillant, c'est à dire une personne qui a le dernier mot sur toutes les décisions prises. Ils pointent facilement les « erreurs » passées de Linus Torvalds (notez les guillemets) : l'utilisation de BitKeeper pour gérer le noyau, l'interdiction de greffons, etc. En tant que développeur logiciel, je pense qu'un dictateur est absolument nécessaire dans un projet de logiciel libre. Voici pourquoi.

Respect gagné par le DBAV

La première raison est aussi certainement la plus importante : le respect. Le dictateur bienveillant à vie (on le nommera DBAV à partir de maintenant) doit prendre des décisions, et même beaucoup de décisions, et en même temps il doit conserver le respect de ses pairs. Les décisions ne sont pas toujours populaires, et elles ne sont pas non plus toujours justes (particulièrement aux yeux des autres). Cependant, le DBAV doit faire preuve de charisme, aussi bien sur le plan technique que relationnel, pour conserver le respect du reste de l'équipe. Personne ne songerait jamais à initier un fork de Linux, car peu de développeurs abandonneraient Linus au profit de la personne à l'origine de la fourche. C'est vrai pour la plupart des projets et c'est pourquoi les forks sont rares. Il faut reconnaître cependant qu'il arrive que le DBAV réussisse à se mettre toute l'équipe de développement à dos, et que quelqu'un finisse par créer une fork en entraînant la majeure partie des développeurs avec lui. Un nouveau projet avec un nouveau nom se crée en s'appuyant sur le code source existant et l'ancien projet disparaît après quelques temps. C'est une bonne chose : le DBAV est en place tant que le peuple le souhaite. C'est une dictature, mais une dictature étrange puisqu'à tout moment les citoyens peuvent s'en aller créer un nouvel état ou en rejoindre un autre.

Connaissance

Le DBAV connaît le projet de A à Z. Il ou elle est à même de savoir si une décision détruira les fondations du projet, et saura également résoudre un problème en conservant la solidité de la structure. Drigg (un projet populaire que je maintiens) m'en apporte la preuve presque chaque semaine : je me rends compte que ma

très bonne connaissance du projet le protège des mauvais patchs (*NdT : correctifs*) et des mauvaises demandes de fonctionnalités. Un DBAV est crucial et il ou elle doit être la personne qui prend les décisions. Dans un logiciel commercial, un manager moyen (qui ne sait pas coder) arrivera parfois à imposer une fonctionnalité ou une modification qui détruira inévitablement la structure du logiciel. Cela m'est aussi arrivé et je pense que c'est arrivé à presque toutes les personnes travaillant sur des logiciels clients propriétaires.

Rapidité

Tout se passe souvent très vite dans le développement de logiciels libres. Parfois les décisions sur la conception et la technique doivent être prises presque dans l'instant. Même si le DBAV peut demander leur avis aux autres membres, c'est à lui ou à elle que revient la décision finale. Il existe une expression anglaise qui dit « A Camel is a horse designed by committee » (*NdT : « Un chameau est un cheval créé par un comité » pour dire que les décisions prises collectivement ne sont jamais optimales*), je la trouve légèrement exagérée, tout dépend évidemment du comité en question, mais c'est en général vrai et c'est bien dommage. Des décisions doivent être prises, et parfois quand on suit la liste de discussion d'un projet, on souhaiterait vraiment que quelqu'un mette fin aux argumentations qui parfois s'éternisent et dise « ok, ça suffit, on va faire les choses comme ça ».

Charge de travail

Soumettre des idées pour de nouvelles fonctionnalités est un droit fondamental (par contre les exiger ne l'est pas). Les membres de l'équipe peuvent débattre de ces demandes de fonctionnalités et de leur implémentation. Cependant, le code fait la loi. Si un utilisateur propose quelque chose qui trouve écho chez une développeur, les discussions peuvent tirer en longueur, mais à un moment « quelque chose doit être codé si vous voulez qu'il se passe quelque chose ». En proposant un patch un membre de l'équipe gagnera du respect et de la crédibilité, à condition bien entendu que le correctif ne détruise pas la structure du projet. Cela signifie donc que les membres qui veulent contribuer prendront en charge ce qui leur tient vraiment à cœur et ils devront coder ces éléments de manière à ce que le DBAV ne les rejettent pas. Cela est bénéfique à la fois pour le code et pour la motivation des gens. Le code créé par les autres membres de l'équipe doit être bon. Ce qui nous amène au point suivant…

Envoyer de bons correctifs

Si un développeur sait que son patch sera inspecté par une personne ayant une connaissance poussée du projet, et que cette personne cherchera la petite bête, il mettra plus d'application dans l'élaboration de son correctif. Si ce n'est pas un DBAV mais un *comité démocratique* qui décide du sort des contributions, alors souvent de mauvais correctifs seront intégrés au code, des correctifs qui seront susceptibles de mettre à mal la structure du projet ou qui auront des effets secondaires obscures et difficiles à corriger. Ceci peut aussi se produire si le DBAV se comporte comme un père de famille confiant et aimant plutôt que comme un vrai DBAV. Oui, il m'est déjà arrivé d'accepter des correctifs trop rapidement, et je me suis retrouvé dans cette situation.

Maintenir la politique à l'écart du projet

Créer un comité c'est ouvrir la porte à la politique. L'équation suivante est souvent vraie : « Politique + Projet technique = Catastrophe ». Si un comité prend toutes les décisions, une partie des membres du comité pourrait finir par s'allier afin de faire passer des décisions pour des raisons autres que techniques (comprendre ici : renvoyer l'ascenseur, demander une faveur, etc.). Le DBAV peut aussi prendre des décisions politiques, mais ce ne seront toujours que les décisions politiques d'une personne, le DBAV lui-même, qui saura qu'une mauvaise décision sur le plan technique sera très préjudiciable au développement du projet.

Pour conclure…

Je ne suis pas spécialement fan des comités. Je pense qu'une méritocratie avec un DBAV fonctionne beaucoup mieux et réalise les choses beaucoup plus rapidement. Durant ma courte expérience de chef de projet de logiciel libre (presque un an), j'ai fait face, à une échelle plus modeste bien sûr, aux mêmes problèmes que rencontrent tous les jours beaucoup de développeurs de logiciels libres. Si vous n'êtes pas d'accord avec mes idées, vous pouvez créer un fork à partir d'un projet et mettre sa gestion dans les mains d'un comité. Mais vous avez de bonne chance d'accoucher d'un chameau plutôt que d'un cheval.

Vous êtes libre de me prouver que j'ai tort.

Notes

[1] Crédit photo : Hoyasmeg (Creative Commons By)