

Microsoft ou les vertus de la monoculture

Pensez-vous par exemple que [la pléthore de distributions GNU/Linux](#) soit une qualité de l'OS et le témoignage de la vivacité de sa communauté ou bien au contraire qu'on aboutit à une situation confuse où *trop de choix tue le choix* ?



Sur cette thématique assez classique de la pertinence de la pluralité du choix, voici la traduction d'un article (un peu technique mais fort intéressant) d'un développeur américain James Turner sur [le site d'O'Reilly](#).

Extrait :

Alors, quels sont les avantages d'une monoculture et pourquoi Microsoft gagne-t-il si souvent quand les gens doivent choisir une plateforme ? C'est en grande partie à cause de ce que la communauté open source voit comme une force mais que ceux qui essaient de faire leur boulot dans le monde réel voient comme une faiblesse. Nous célébrons la diversité de choix disponibles pour résoudre un problème et nous appelons cela la liberté. Les directeurs informatiques et les patrons de la branche informatique (IT managers et CIOs en anglais) y voient du chaos, de la confusion et des doutes.

Pour ceux qui comme nous sont attachés à la liberté, avoir le choix est bien entendu une valeur fondamentale. Mais il peut en aller autrement dans le monde pragmatique de l'informatique professionnelle où c'est souvent l'efficacité qui est privilégié. Et alors dans ce contexte Microsoft conserve de sérieux atouts avec ses offres *monoculturelles* sécurés et

rassurantes^[1].

Les vertus de la monoculture

[The Virtues of Monoculture](#)

*James Turner – 24 avril 2007 – Opinion
(Traduction Framalang : Don Rico et Yostral)*

Je ne dis certainement rien de nouveau ici, mais j'ai pensé que je pourrai partager quelques réflexions sur les raisons qui poussent les gens à suivre la voie Microsoft. J'ai récemment fait quelque chose dans mon travail *de tous les jours* auquel je pensais depuis longtemps, mais pour lequel je n'ai jamais vraiment pris la peine d'aller jusqu'au bout, je me suis inscrit pour participer à un projet Microsoft-centrique et pour apprendre le .NET.

J'avais fait des tentatives avortées par le passé pour apprendre à coder dans l'Univers Microsoft. J'avais fait un essai à la sale époque des COM, mais le nombre de numéros qu'on me demandait d'exécuter me demandait trop d'effort par rapport à ce que j'étais alors prêt à consentir. Depuis j'ai gardé ce mauvais goût au fond de la bouche et j'ai refusé d'ajouter une seule compétence Microsoft à mon répertoire, même si cela représentait parfois un vide dans mon curriculum vitæ.

J'ai souvent travaillé dans des environnements où il y avait ce Monsieur Microsoft, l'évangéliste qui vous répète sans cesse à quel point ça aurait été plus facile en .NET. Je les ai classés dans la catégorie adoreurs de Gates buveurs de Tang*. Mais, à la fin de la journée, je me suis dit que si je devais les critiquer je devais vraiment comprendre leur monde. Connais ton ennemi et tout ça.

J'ai passé la semaine dernière à apprendre dans l'ordre C#, .NET et VSTO (c'est Visual Studio Toolkit for Office, si les abréviations de Microsoft ne sont pas votre tasse de thé).

J'ai utilisé le livre *Learning C#* de chez O'Reilly et j'ai fait quelque chose qui m'arrive rarement : je m'y suis mis de manière très méthodique (du moins pour la première moitié).

Et devinez quoi? Microsoft possède dans ses mains une suite de développement plutôt bonne. Pour être honnête, C# est vraiment ce que je ferai si je pouvais complètement ré-écrire Java sans me soucier de la compatibilité descendante. Il y a quelques fonctionnalités vraiment sympas, comme les mots-clés `virtual`, `override`, et `new` qui vous permettent de spécifier ce qu'il se passe lorsque vous transtypez une classe dans sa classe de base et que vous appelez une méthode qui est définie dans les deux.

Visual Studio est un outil habile qui vous permet vraiment de créer des applications (et avec VSTO des ajouts pour Office) en deux temps trois mouvements. ADO.NET n'est pas pire que JDBC et s'intègre de manière transparente dans Visual Studio. J'ai été capable, arrivé à la fin de la semaine, de développer des applications autonomes et des ajouts pour Office qui étaient capable de dialoguer avec les bases de données en n'ayant écrit que peu de code. D'après ce que j'en ai vu, ASP.NET réalise la même chose pour les applications web MVC (*NdT : Model View Controller*).

Alors, quels sont les avantages d'une monoculture et pourquoi Microsoft gagne-t-il si souvent quand les gens doivent choisir une plateforme ? C'est en grande partie à cause de ce que la communauté open source voit comme une force mais que ceux qui essaient de faire leur boulot dans le monde réel voient comme une faiblesse. Nous célébrons la diversité de choix disponibles pour résoudre un problème et nous appelons cela la liberté. Les directeurs informatiques et les patrons de la branche informatique (IT managers et CIOs en anglais) y voient du chaos, de la confusion et des doutes.

Est-ce que je devrais utiliser iBatis ou Hibernate? XFire ou AXIS? Perl, PHP ou Ruby? Debian, Fedora, Ubuntu ou Suse? Si

vous prenez la mauvaise décision vous pouvez perdre énormément de temps, comme nous l'avons découvert sur un projet récent où nous avons gâché une semaine à essayer de faire marcher AXIS2 pour un projet de service web pour finalement nous rendre compte que XFire était ce qu'il nous fallait.

Pour Monsieur Microsoft cette confusion n'existe pas. Vous utilisez ADO.NET, ASP.NET, C# et Windows. Ils fonctionnent tous, ils sont tous bien documentés du point de vue des besoins des développeurs, sans un seul *regarde le code source* désobligeant. A chaque fois que je pensais que j'allais être bloqué il y avait une douzaine d'articles expliquant comment faire exactement ce que je voulais faire, avec un exemple de code qui était à jour avec les versions du logiciel que j'utilisais et qui répondait vraiment au problème que je cherchais à résoudre.

Microsoft apporte le confort de ne pas avoir à choisir. Avoir le choix n'est pas toujours bon et la communauté open source offre parfois bien trop de manières différentes de plumer un canard, des choix qui sont pris plus par fierté, ego ou entêtement que par une authentique nécessité d'avoir deux alternatives différentes. Je ne montrerai personne du doigt, tout le monde connaît des exemples.

En fait, à moins que vous ne pensiez que je me sois tourné vers le Côté Obscur, le GROS problème avec une monoculture, c'est que vous vendez plus ou moins votre âme pour la stabilité d'un ensemble de choix défriché pour vous. En empruntant le chemin .NET, en gros, vous vous y perdez à tout jamais, et ce malgré Mono. Vous travaillerez toujours sur une plateforme Windows. Vous avez le joli anneau en or, mais Sauron tire les ficelles et vous fait danser. Pour beaucoup d'entreprises, celles qui n'ont pas besoin de se soucier du déploiement dans un environnement hétérogène, c'est un pacte qu'elles sont plus que prêtes à conclure.

Voici ce que je retiens de toute cette réflexion : en quelque

sorte, nous devons commencer à faire le tri. La massue de 350kg pour faire entrer certaines idées dans les têtes devrait être mise à disposition pour marteler les têtes de ceux qui fourchent (*NdT : qui créent un [fork](#) une déviation indépendante d'un projet*) pour la seule et unique raison qu'ils ne sont pas en accord avec la licence, ou de ceux qui prennent les décisions. Quand on entend parler de deux (ou plus) projets qui répondent à la même problématique, on devrait se demander « Pourquoi ne mettent-ils pas en commun leurs efforts pour fournir une très bonne solution? » plutôt que de célébrer la diversité uniquement pour l'amour de la diversité.

A-ton vraiment besoin de Ruby on Rails ET de Groovy on Grails? Quand ils ont annoncé le poisson d'avril de Python on Planes j'ai mis quelques secondes pour réaliser que c'était un canular, parce que c'est exactement le genre d'effort faire quelque chose pour l'amour de le faire qui fractionne la communauté des logiciels open source. Il n'y a aucun moyen d'empêcher les gens de commencer des projets en double, et nous ne le voudrions pas, mais bon sang, doit-on l'encourager activement ?

On passe beaucoup de temps à se plaindre des moyens démoniaques qu'emploie Microsoft pour s'imposer partout. En faisant cela, nous nous lavons automatiquement de toute responsabilité que nous pourrions nous-même porter pour leur succès ou nos échecs. Le fait est qu'il existe d'excellentes raisons pratiques qui poussent les gens dans les bras de la boîte à outil de Redmond et nous devons accepter ceci comme un fait et en tirer des leçons plutôt que d'agiter nos poings en blamant l'obscurantisme. Car nous avons trouvé notre ennemi et c'est nous, pas Microsoft, du moins pas tout le temps...

Notes

[1] Crédit photo : [Zach Klein](#) (Creative Commons By)