

---

# Motiver ses développeurs

Forum PHP Paris 2009

---



On veut que les développeurs soient productifs et on veut les garder dans l'entreprise. Pour ce faire, il faut comprendre leurs besoins et trouver le moyen de les motiver.

Chefs d'équipe? Chargés de projets? Autre poste qui implique la motivation des développeurs?

Débat à la fin.

# Qui suis-je?

- Anna Filina
- Développement Web et autre depuis 1997
- Organisation de conférences Web depuis 2006
- Motive les développeurs de ConFoo.ca quotidiennement

Les informations présentées viennent de mes observations, de mes discussions avec des développeurs et des gestionnaires, et de différentes études publiées au Québec.

ConFoo: 12 organisateurs, pour la plupart des développeurs, venant de communautés différentes: PHP, Python, Ruby, ... Les organisateurs étant tous des bénévoles, il n'est pas toujours facile de les motiver. C'est mon travail.

# Plan

- La problématique
- Trousse de survie
- Comprendre le développeur
  - Débutant
  - Expert
- Phobies

# La problématique

- Difficile à recruter
- Manque de coopération
- Avancent trop lentement
- Quittent sans donner de raison



Développeur tel que vu par un non développeur:

- Il n'est pas facile de l'attirer dans l'entreprise. Il y a trop de demande et pas assez d'experts.
- Il est souvent en désaccord avec les tâches qu'on lui assigne. Il ne fait pas les tâches selon l'ordre des priorités. Il ignore complètement certaines directives, comme envoyer un rapport en fin de journée.
- Il avance trop lentement par rapport aux attentes. Cela engendre des retards et les clients sont mécontents.
- Il arrive souvent qu'un développeur quitte l'entreprise. Les raisons pour cela sont généralement floues, et le gestionnaire ne sait pas comment éviter d'autres départs dans le futur.

# Trousse de survie

- Bon salaire
- Outils adaptés
- Prendre un verre
- Formation payée (comme le Forum PHP ou ConFoo!)



- Le salaire est le critère le plus important selon une étude publiée dans le magazine JobBoom au Québec.
- Il est important de leur fournir les bons outils pour être productif. Pouvoir choisir les outils aide également.
- Prendre un verre un soir de temps en temps est bien pour aider les développeurs à décontracter. Il faut encourager les employés à organiser ça pour ne pas avoir le sentiment que c'est obligatoire.
- Les formations payées sont un très bon moyen de se faire aimer par ses développeurs.

# Comprendre le débutant

- Fait ses preuves, ne pose pas de questions
- Écrit du code
- Besoin d'une liste de tâches à faire
- Besoin de validation



- Il sent qu'il est constamment testé et veut bien paraître. A cause de ça, il n'ira pas poser de questions aux développeurs plus expérimentés par crainte qu'on lui dise qu'il n'est pas très bon.
  - vérifier régulièrement son progrès pour éviter qu'il ne soit bloqué
  - l'inviter à poser des questions, même si elles sont simples
  - bien lui montrer et expliquer les outils qu'il ne connaît pas
- Le débutant est là pour coder. Il n'a pas les compétences pour planifier ou résoudre des problèmes sur un projet.  
"Voilà les spécifications, occupes-toi de faire le projet du début à la fin"
  - lui donner des tâches qu'il est capable d'accomplir
  - l'aider dans les décisions
  - le jumeler à un expert qui aura du temps pour répondre à ses questions
- Il ne sait pas ce qu'il y a à faire sauf si on le lui dit. Il n'a pas de vision globale du projet.  
"Fais ce qu'il reste à faire"
  - donner une liste claire de tâches à faire
  - lui dire dans quel ordre faire les tâches s'il y a des dépendances
- Il ne peut pas travailler longtemps sans qu'on valide son travail.
  - l'informer rapidement si ce qu'il fait est bon ou n'est pas bon
  - une tape sur l'épaule de temps en temps fait du bien

# Comprendre l'expert

- Fait avancer le projet
- Éteint les feux
- A besoin d'outils
- Micro-gestion



- L'expert sait ce qu'il doit faire pour terminer le projet.
  - lui donner une grande autonomie
  - mettre un débutant ou un intermédiaire à sa disposition pour la possibiler de déléguer des tâches plus simples; cela augmentera sa productivité et évitera des bloquages dans le projet
  - lui permettre de communiquer directement avec le client quand cela est nécessaire
- Il est capable d'éteindre les feux, c'est-à-dire se débrouiller lors des retards de livraisons ainsi que de réparer les bogues urgents. Par contre, il ne faut pas en abuser.
  - ne pas l'utiliser uniquement pour éteindre des feux car ce rythme va l'épuiser et l'emmerder
  - bien planifier l'échéancier afin d'éviter les situations où il faut courir
- Il a besoin d'avoir des outils adaptés.
  - 2 écrans, chaises confortables
  - aujourd'hui, un système de contrôle de révision est indispensable, même s'il n'y a qu'un seul développeur sur le projet
    - si possible, accorder assez de temps pour les tests unitaires; on ne veut pas que le développeur aie peur de jouer dans le code car il peut briser du code qui marche déjà
    - avoir plusieurs environnements (minimum développement, test et production) car le développeur veut développer à l'aise mais ne veut pas être responsable de bogues en production
- Il n'aime pas qu'on regarde par dessus son épaule quand il travaille.
  - lui laisser une certaine flexibilité dans la gestion de son temps
  - lui laisser une grande liberté dans les décisions techniques
  - garder les rapports et les réunions au minimum

# Phobies

- Objectifs pas clairs
- Changements de spécifications
- Délais irréalistes
- Temps supplémentaire
- Activités en fin de semaine

- Un développeur ne peut travailler sans objectifs clairs et mesurables. Il doit avoir des tâches précises et savoir pour quand il doit les terminer.
- Les changements de spécifications sont une réalité, mais il ne faut pas les faire en cours d'un sprint (scrum). Si des spécifications changent au cours du projet, il faut bien les communiquer au développeur.
- Il y a des délais courts et il y a des délais irréalistes. Le client ou le gestionnaire ne sait pas toujours combien de temps peut prendre une tâche. Ce qu'il croit être une affaire de quelques minutes implique en fait la restructuration du code et plus de 20 heures de travail. Toujours consulter le développeur avant de faire une promesse à un client.
- Le temps supplémentaire épuise le développeur. Cela peut mener rapidement à son départ de l'entreprise. Il faut l'éviter autant que possible en laissant une marge de manoeuvre sur un projet.
- Les activités comme le paintball ou la randonnée sont intéressantes, mais en fin de semaine le développeur veut passer du temps en famille ou avec ses amis. S'il faut faire une activité qui renforce l'esprit d'équipe, faites-les sur les heures de travail. Ça augmente tellement la productivité, que l'investissement en vaut la peine.

# Gadgets

- Insigne magnétique
- Nerf Gun
- Jeux vidéo sur grand écran
- Quelques idées sur [thinkgeek.com](http://thinkgeek.com)



# Me rejoindre

- @afilina
- annafilina.com
- [afilina@confoo.ca](mailto:afilina@confoo.ca)