

Paris on Rails: 'Devenez adultes en matière de développement'

Laurent Juilliard et Richard Piacentini sont les fondateurs et directeurs associés de Nuxos Group. Après vingt ans d'activité dans l'informatique, ils ont découvert Ruby on Rails et s'affichent aujourd'hui comme les évangélistes de cette plate-forme de développement en France.

Ruby on Rails est différent des frameworks de programmations que l'on a l'habitude de rencontrer, les PHP, Java et autres .NET. Rails est bourré de conventions qui imposent des contraintes au développeur, et paradoxalement ce sont ces contraintes qui en font sa force et le rendent très pratique pour les développeurs !

« Rails assure une partie du travail. Il définit des cadres et remet à plat notre façon de programmer en nous permettant d'éviter la cacophonie que l'on obtient habituellement en empilant des composants », nous confient les deux compères.

En fait, Rails est battu au dessus de Ruby, un langage qui permet de construire des langages spécialisés personnalisés par domaine d'application, tout en offrant une intégration complète de tous les composants.

« On personnalise pour simplifier. Un seul langage suffit pour maîtriser toute la pile applicative. Rails intègre tous les composants, mais c'est aussi un framework qui implémente MVC[Model-View-Controller, une architecture et une méthode de conception reconnue] qui repose sur le pattern MVC. »

« A la différence des autres langages, il impose une séparation des parties de l'application. Et toutes les applications Rails sont architecturées sur le même modèle, ce qui en facilite l'accès pour les programmeurs. »

Rails se présente donc comme un environnement de développement et de programmation d'applications Web, ce qui le différencie également d'applications du type Dreamweaver.

« Rails ne s'adresse pas aux mêmes utilisateurs, il est destiné au développement des applications Web, comme l'e-commerce, le reporting, les applications dynamiques. Les conventions imposent de se concentrer sur le développement. Elles permettent également de ne pas réécrire toujours la même chose. »

« C'est un avantage pour le chef de projet. Il dispose d'un cadre cohérent et offre de sérieux gains de temps sur la quantité de code produit. La complexité métier est identique à celle des autres solutions de développement, mais il faut quatre fois moins de code pour faire la même chose ! »

« Nous devons faire face à une constante : le nombre de bugs est proportionnel au nombre de lignes de code. Avec Ruby on Rails, nous avons inséré notre savoir-faire dans le framework. Du coup, on ne retrouve pas dans Rails la complexité accidentelle induite par Java, par exemple. »

A l'inverse d'autres acteurs du marché, Ruby on Rails est une solution open source, mais ne s'en vante pas !

« Cela nous permet de créer une dynamique et de multiplier les contributions. Mais ce n'est jamais un

argument mis en avant ! Rails préfère mettre en avant les bonnes pratiques, par exemple sur le test de code. Au fur et à mesure de la programmation, il développe automatiquement un squelette de tests. Le programmeur ne se retrouve donc pas devant une page blanche ! Ce n'est pas une option, mais on ne trouve pas cela chez les autres éditeurs où c'est toujours une annexe.»