***Note : le texte est la méthodologie utilisée dans la boite d’Eric. Le dessin a été rayé de rouge pour mieux correspondre à leur méthodologie mais il faudrait le retrouver sans ratures (cf lien plus bas).***

On a un workflow à base de branch de **features** et branch de **maintenance** autour de la branch **master**

Quand on fait un développement, refactoring, ou mise à jours d'une bibliothèque, on le fait dans une feature branch spécifique qui devrait partir de la branch master ou d’integration (development branch). Cette branch peut-être testé sur différentes platform grâce aux buildbots, et mergée dans master.

Quand on fait une release, on fait un setup dans master tant qu'on a pas développé toutes les features du setup. Durant cette période on fait attention à ne merger que des choses sans risque ou nécessaire à la release dans master.   
Une fois qu'on a un setup que l'on pense définitif on peut créer une branch de maintenance qui part de master : Elle sert aux derniers bugfix.

Notre workflow est proche du fameaux ["successful git branching model"](http://nvie.com/posts/a-successful-git-branching-model/) : on utilise la branch master comme branch develop, ce qu'ils appellent release branches on appelle ca branch de maintenance, et on n'a pas l'équivalent de leur branch master car on a plusieurs produits.

