

# Introduction à git

Olivier Lafleur

4 mai 2015

## Table des matières

|          |                                |          |
|----------|--------------------------------|----------|
| <b>1</b> | <b>Introduction</b>            | <b>1</b> |
| <b>2</b> | <b>Pourquoi utiliser git ?</b> | <b>1</b> |
| <b>3</b> | <b>Auteur</b>                  | <b>2</b> |

## 1 Introduction

Ce document se veut une introduction au système de gestion de versions décentralisé [git](#). Nous utiliserons l'exemple de git, mais les principes présentés pourraient aussi bien être appliqués à d'autres systèmes du genre comme Mercurial ou BitKeeper.

## 2 Pourquoi utiliser git ?

Lorsque l'on développe du code, il n'est pas rare que l'on soit plusieurs à vouloir travailler sur un même bout de code. Cependant, cela peut devenir problématique lorsque l'on veut changer des bouts de code à des endroits différents.

En utilisant le courriel ou une clé USB on peut s'échanger des bouts de code, mais cela devient rapidement très complexe et il est facile de faire des erreurs lors de l'intégration du code d'un collègue dans le sien.

Une technique est de travailler chacun son tour : la personne A commence le code, envoie le résultat à une personne B, qui ajoute un autre bout de code et renvoie le tout à la personne A. Ce va-et-vient peut être laborieux et ralentir de beaucoup ce qu'il serait possible de réaliser en travaillant en parallèle.

Quels seraient donc les avantages d'utiliser la gestion de versions ? Cela est mieux puisque :

- Rien qui est sauvegardé (on dit faire un *commit*) n'est perdu. Cela veut dire que l'on peut l'utiliser comme la fonctionnalité pour revenir en arrière (*undo*) d'un éditeur. Par ailleurs, puisque toutes les anciennes versions sont sauvegardées, il est toujours possible de revenir dans le temps et de se replacer dans un état passé.
- Nous avons une liste des changements qui ont été faits, par qui et quand. On sait donc à qui poser nos questions plus tard.
- C'est difficile (mais pas impossible) de réécrire par-dessus les changements de quelqu'un. Le système de gestion de versions avertit automatiquement l'utilisateur lorsqu'il y a un conflit entre deux changements effectués sur la même ligne.

## 3 Auteur

Olivier Lafleur ([olivier.lafleur@c11.qc.ca](mailto:olivier.lafleur@c11.qc.ca))

Ce contenu est lui aussi mis à votre disposition selon les termes de la [Licence Creative Commons Attribution 4.0 International](#) CC-BY.

Par ailleurs, il est important de mentionner que le contenu de ce document est fortement inspiré du contenu des formations de l'organisme [Software Carpentry](#), organisme à but non lucratif qui met disponible en ligne son contenu sous licence Creative Commons.