

# Git, Essayons de reprendre le contrôle !

Antoine de ROQUEMAUREL

 satenske

Développeur Java consultant chez Tech Advantage



Meetup Java / C# du 28 Mars 2019

# LE LOGICIEL DE GESTION DE VERSIONS



Have you ever:

- Made a change to code, realised it was a mistake and wanted to revert back?
- Lost code or had a backup that was too old?
- Had to maintain multiple versions of a product?
- Wanted to see the difference between two (or more) versions of your code?
- Wanted to prove that a particular change broke or fixed a piece of code?
- Wanted to review the history of some code?
- Wanted to submit a change to someone else's code?
- Wanted to share your code, or let other people work on your code?
- Wanted to see how much work is being done, and where, when and by whom?
- Wanted to experiment with a new feature without interfering with working code?

In these cases, and no doubt others, a version control system should make your life easier.

To misquote a friend: A civilised tool for a civilised age.

[share](#) [improve this answer](#)

[edited Nov 6 '13 at 0:52](#)

[answered Sep 11 '09 at 0:42](#)



[si618](#)

14.4k

12

60

79

FIGURE – Pourquoi devrais-je utiliser le contrôle de version ?<sup>1</sup>

---

1. <https://stackoverflow.com/questions/1408450/why-should-i-use-version-control>

# GIT



- ▶ Créé en 2005 par Linus Torvalds
- ▶ Décentralisé
- ▶ Excellente gestion des branches
- ▶ Efficace sur de gros projet

# GIT



- ▶ Créé en 2005 par Linus Torvalds
- ▶ Décentralisé
- ▶ Excellente gestion des branches
- ▶ Efficace sur de gros projet
  - ▶ Microsoft Windows :
    - ▶ 3 500 000 fichiers, soit 300 Go
    - ▶ 440 branches
    - ▶ 4 000 utilisateurs
    - ▶ 10 000 merges

# GIT



- ▶ Créé en 2005 par Linus Torvalds
- ▶ Décentralisé
- ▶ Excellente gestion des branches
- ▶ Efficace sur de gros projet

*« I'm an egotistical bastard, and I name all my projects after myself.  
First 'Linux', now 'git'. »*

## Les bases

## Le workflow Git

## La collaboration

# LE SYSTÈME DÉCENTRALISÉ

# LA ZONE DE TRANSIT (*staging area*)

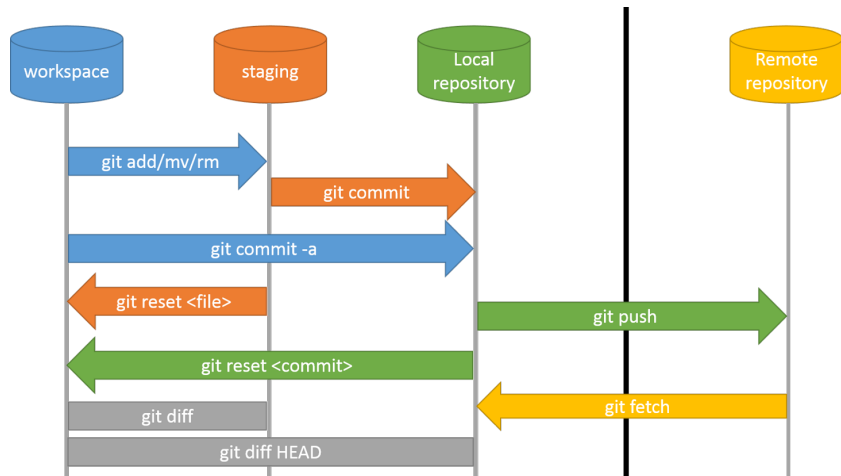


FIGURE – Fonctionnement de Git



# LES ACTIONS DE BASE

- ▶ `commit`
- ▶ `fetch`
- ▶ `push`
- ▶ `blame`

Les bases

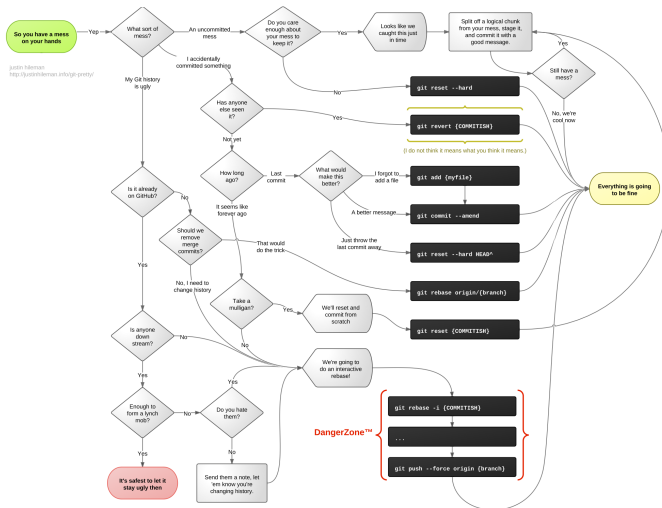
Le workflow Git

La collaboration

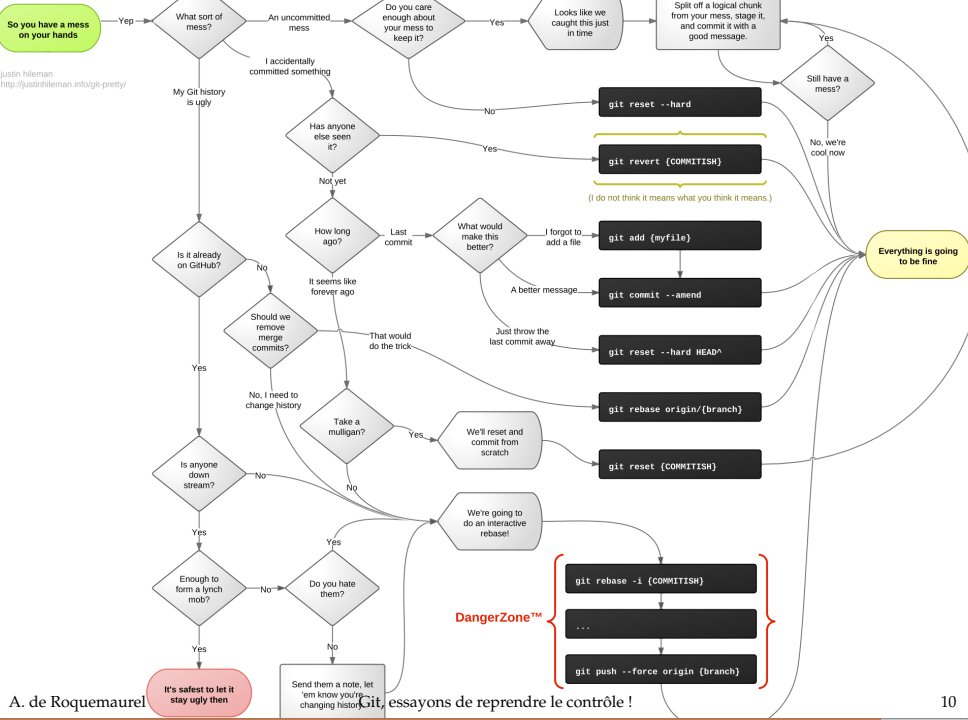
Les bases

Le workflow Git

La collaboration

FIGURE – So, you have a mess on your hands? <sup>2</sup>

2. <http://justinhileman.info/article/git-pretty/>



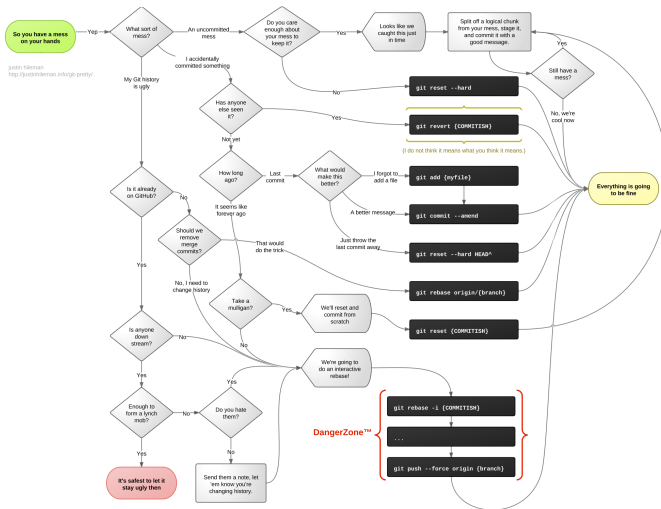
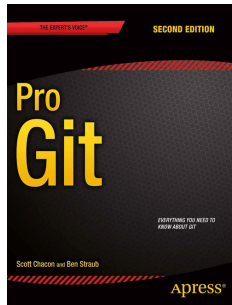
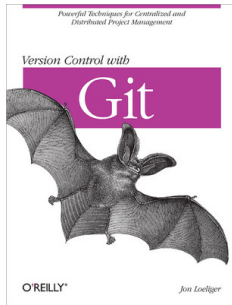


FIGURE – So, you have a mess on your hands? <sup>2</sup>

2. <http://justinhileman.info/article/git-pretty/>

# RÉFÉRENCES



- ▶ <https://git-scm.com>  
Site officiel
- ▶ <https://learngitbranching.js.org>  
Apprendre Git de manière ludique