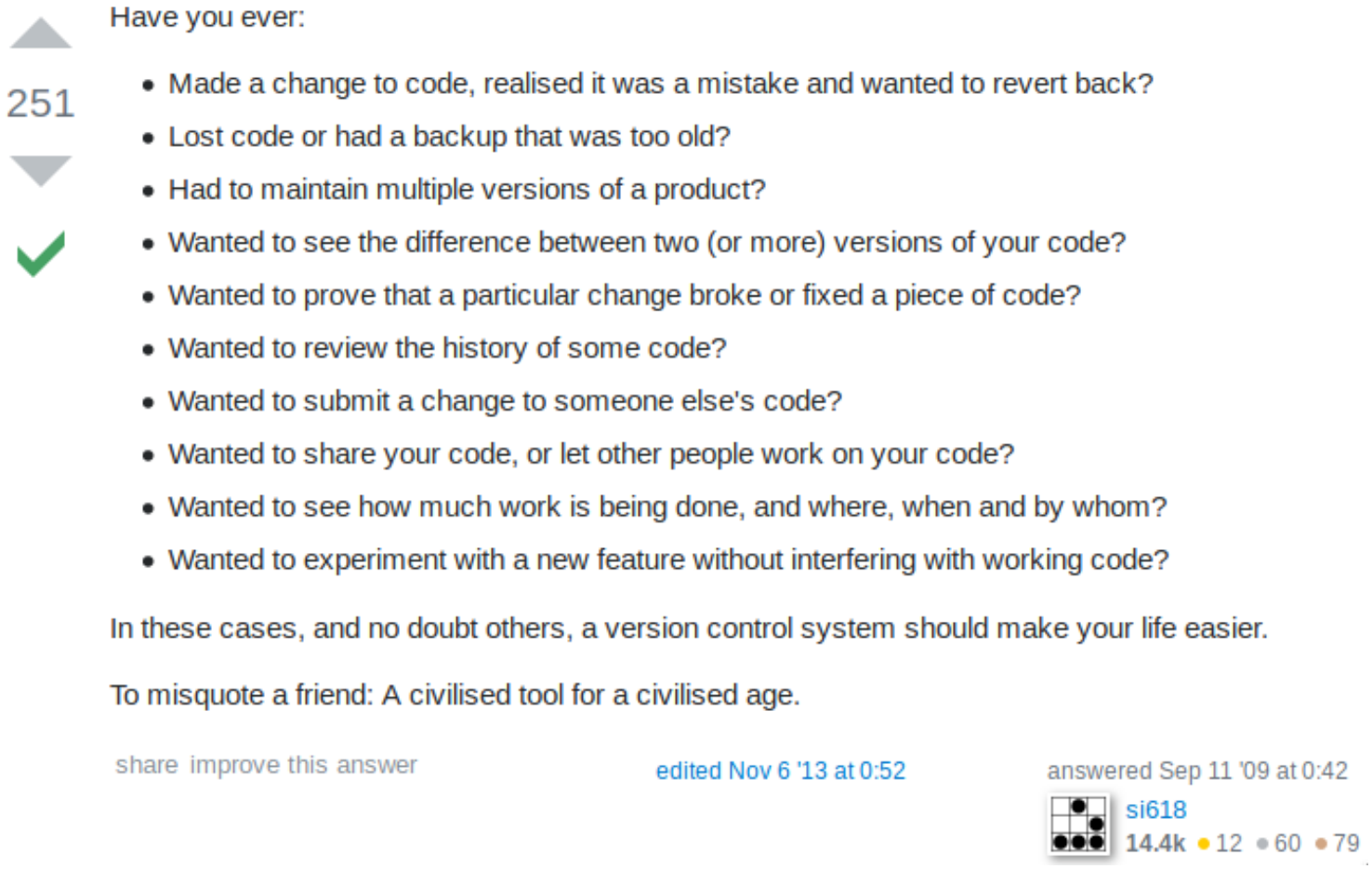
Git, Essayons de reprendre le contrôle !

Antoine de ROQUEMAUREL
satenske

Développeur Java consultant chez Tech Advantage

Meetup Java / C# du 28 Mars 2019

LE LOGICIEL DE GESTION DE VERSIONS



251

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 ? [1]

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*)

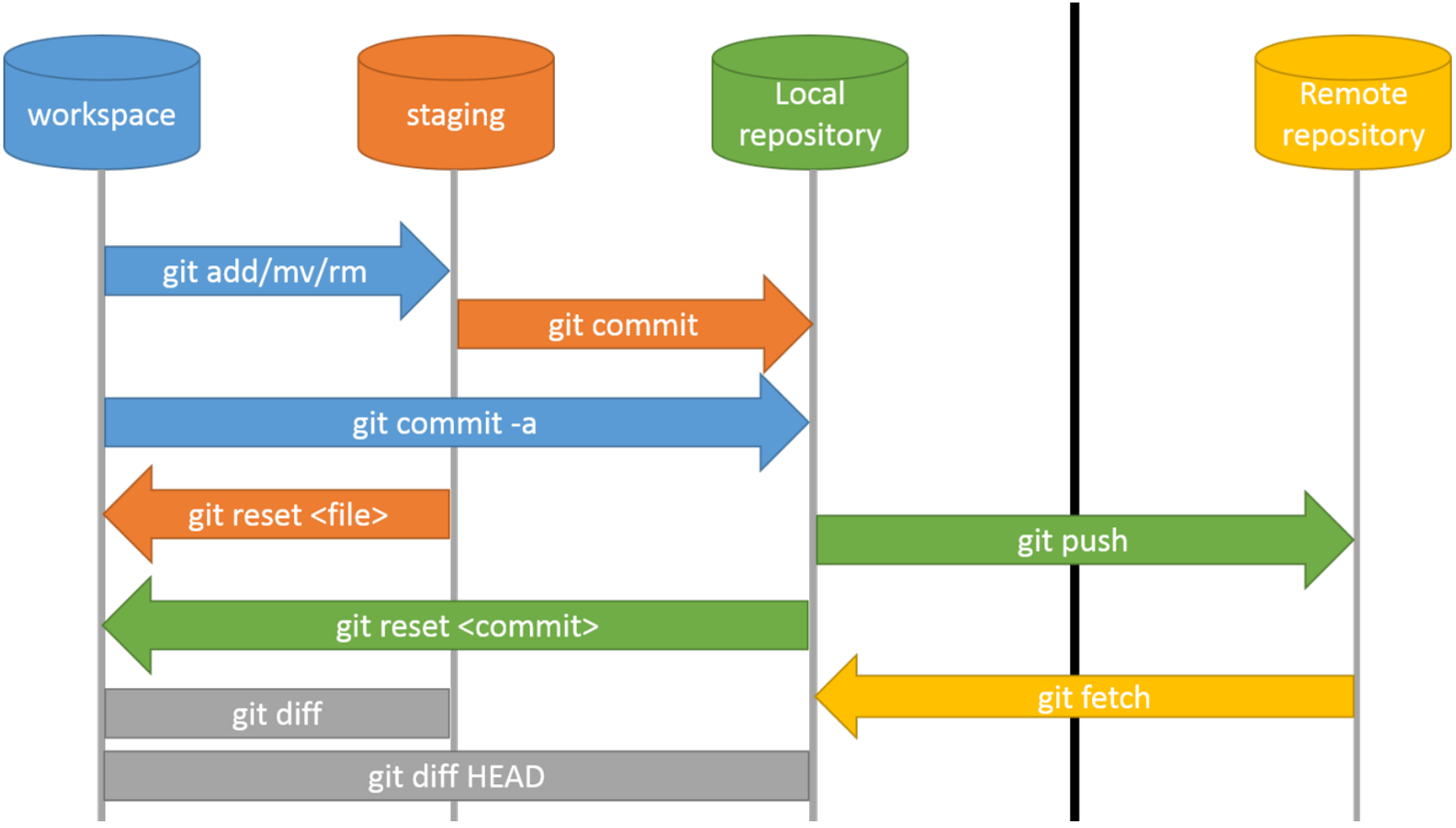


FIGURE – Fonctionnement de Git

Les actions de base

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

Les bases

Le workflow Git

La collaboration

Les bases

Le workflow Git

La collaboration

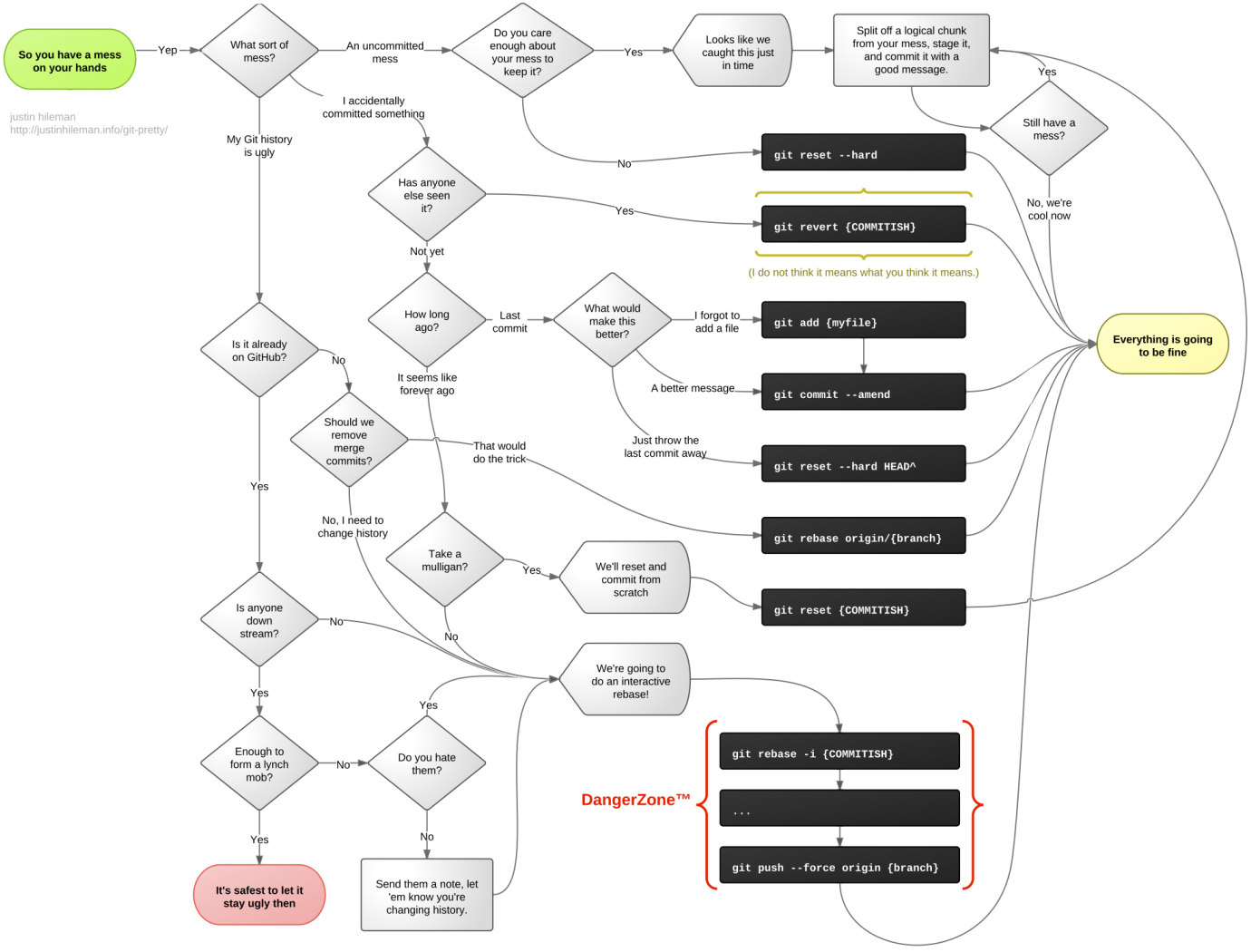
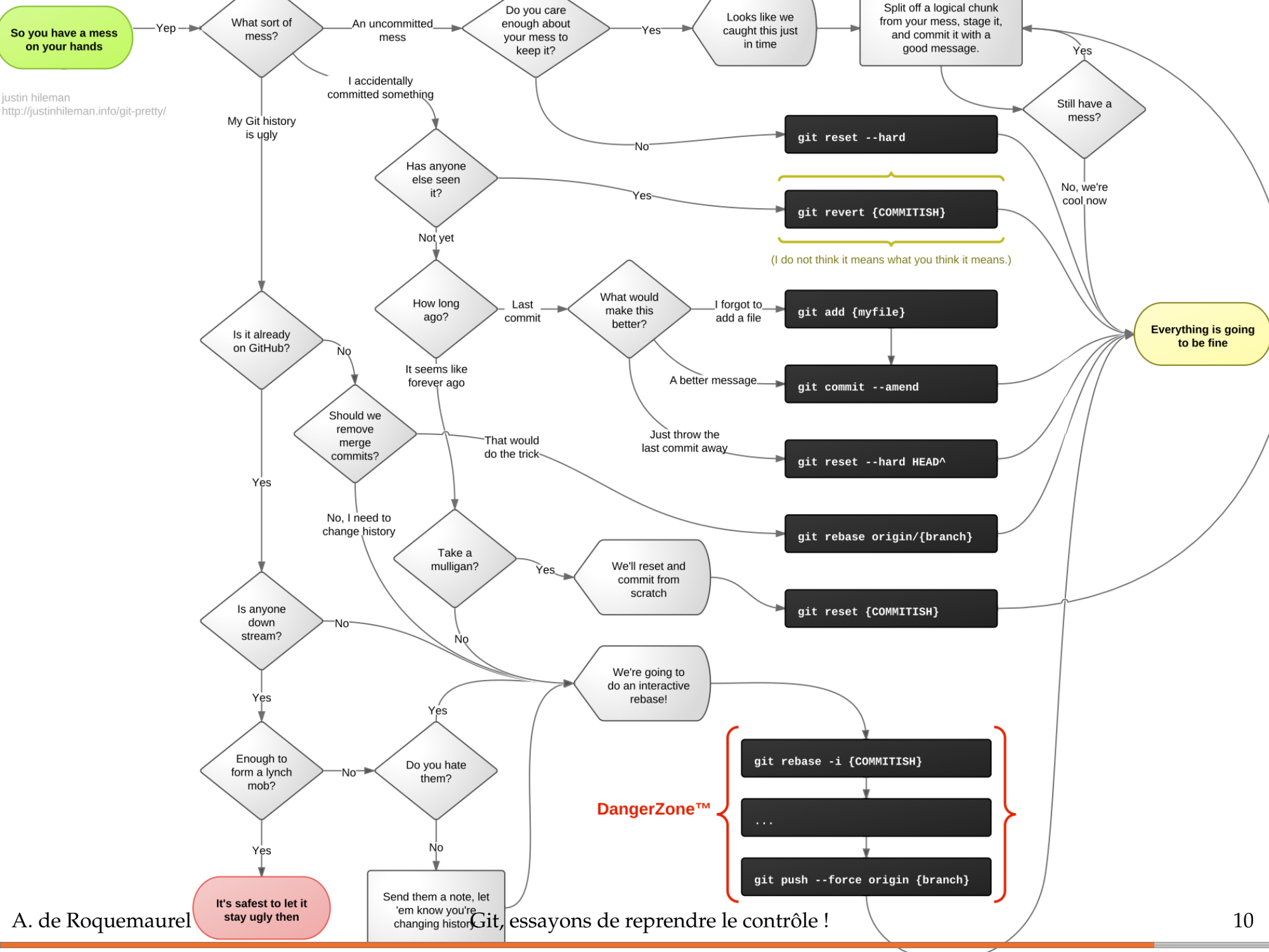


FIGURE – *So, you have a mess on your hands ?* [2]

2. `http://justinhileman.info/article/git-pretty/`


So you have a mess on your hands
Yep
What sort of mess?
An uncommitted mess
Do you care enough about your mess to keep it?
Yes
Looks like we caught this just in time
Split off a logical chunk from your mess, stage it, and commit it with a good message.
Yes
Still have a mess?
No, we're cool now
No
git reset --hard
I accidentally committed something
My Git history is ugly
Has anyone else seen it?
Yes
git revert {COMMITISH}
(I do not think it means what you think it means.)
Not yet
How long ago?
Last commit
What would make this better?
I forgot to add a file
git add {myfile}
A better message
git commit --amend
Just throw the last commit away
git reset --hard HEAD^
It seems like forever ago
Is it already on GitHub?
No
Should we remove merge commits?
That would do the trick
git rebase origin/{branch}
Yes
No, I need to change history
Take a mulligan?
Yes
We'll reset and commit from scratch
git reset {COMMITISH}
No
Is anyone down stream?
No
We're going to do an interactive rebase!
Yes
Yes
Enough to form a lynch mob?
No
Do you hate them?
Yes
No
DangerZone™
git rebase -i {COMMITISH}
...
git push --force origin {branch}
It's safest to let it stay ugly then
Send them a note, let 'em know you're changing history
Everything is going to be fine
justin hileman
http://justinhileman.info/git-pretty/

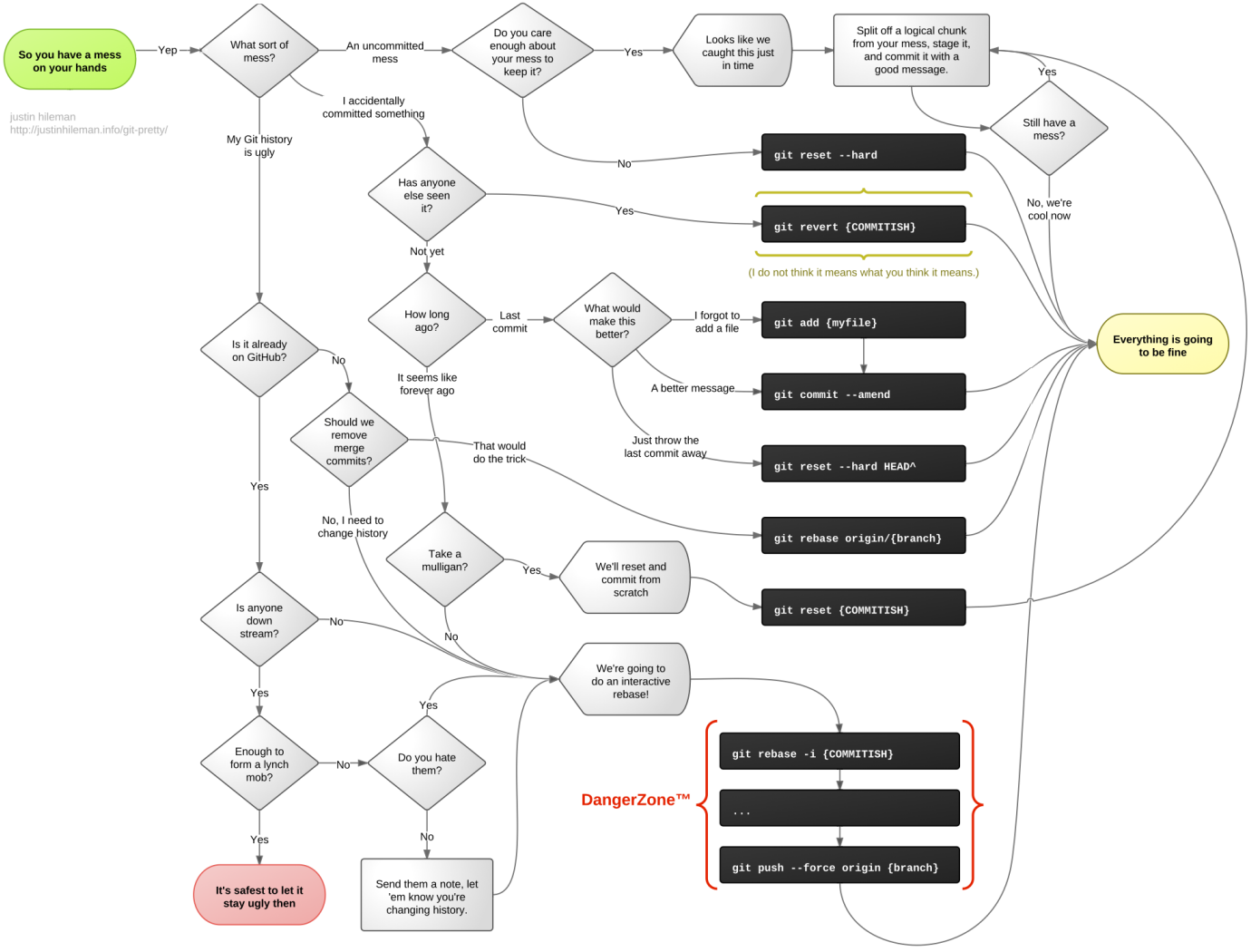
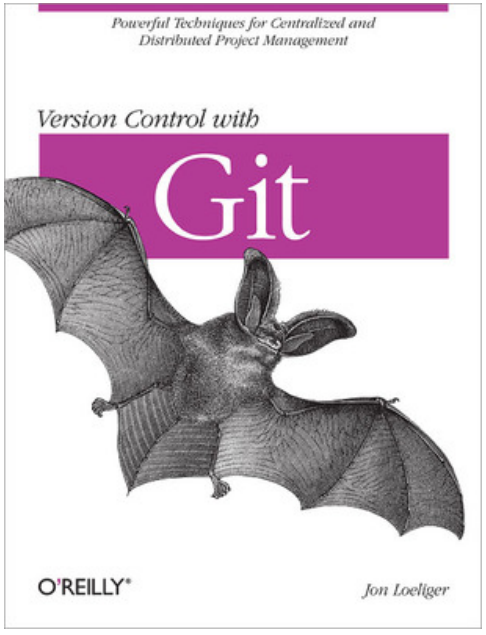
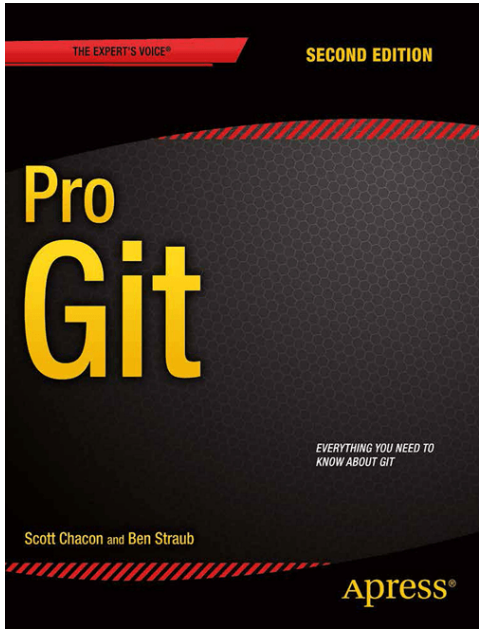


FIGURE – *So, you have a mess on your hands ?* [2]

2. `http://justinhileman.info/article/git-pretty/`

Références





- `git-scm.com`
  Site officiel
- `learngitbranching.js.org`
  Apprendre Git de manière ludique
- `github.com/aroquemaurel/Presentation-beamer-Git`
  Les sources LaTeXde cette présentation