

m04r productivity !

Hugo Mougard

29 novembre 2013

Outline

- 1 Le souci
- 2 Git
- 3 Les Autotools
- 4 Maven
- 5 Autres outils

L'approche naïve n'est pas la bonne

Il était une fois l'étudiant naïf de licence en Informatique et son prof, qui lui donna un projet à réaliser avec 3 de ses potes...

Gestion du code

L'idée naïve

Chacun gère de son côté et on partage par mails à tout le groupe.

Problèmes

- conflits quand plusieurs personnes codent sur le même fichier
- très dur de retrouver l'état du code à un moment donné
- gestion des versions par archives assez vite bordélique

Gestion du build

L'idée naïve

Un projet Eclipse par personne, et utilisation du joli bouton vert pour lancer le programme

Problèmes

- gestion manuelle des dépendences
- difficulté à importer le travail des autres
- tests non automatiques
- ...

Rescue time





Kezako ?

Logiciel qui sert à :

- versionner son code
- combiner son travail à celui des autres
- combiner son travail à... son travail
- partager son code
- chasser les bugs
- avoir un historique de développement

Pour apprendre

<http://git-scm.com/book/> chapitres 1 & 2 et
<http://try.github.io>

Sous emacs

Coucou, tu veux voir Magit ?

Les bases, en images, 1/7

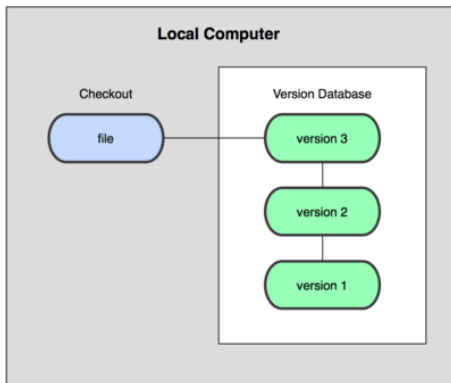


FIG. : Un système de gestion de code « basique »

Les bases, en images, 2/7

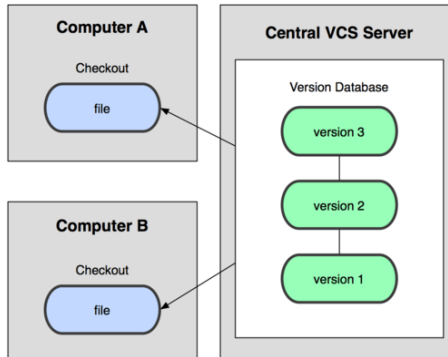


FIG. : Un système de gestion de code centralisé

Les bases, en images, 3/7

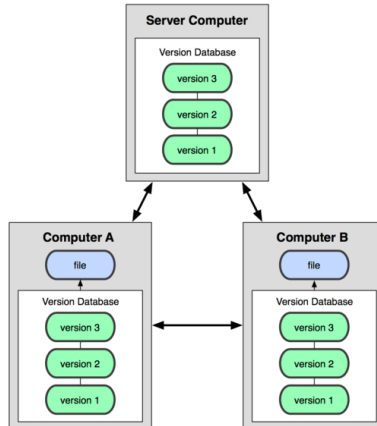


FIG. : Un système de gestion de code décentralisé

Les bases, en images, 4/7

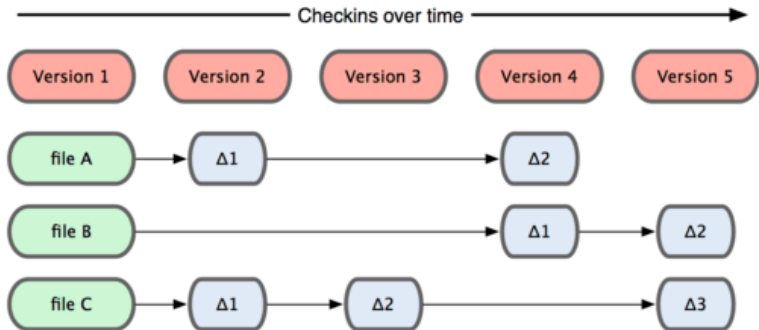


FIG. : La base de données est en fait un truc pas crade

Les bases, en images, 5/7

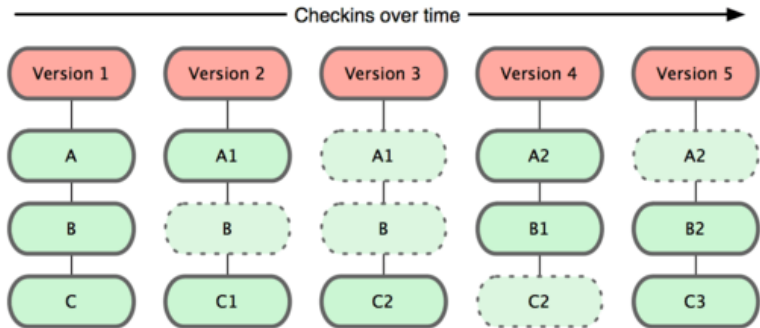


FIG. : Qui permet quand même de tout retrouver

Les bases, en images, 6/7

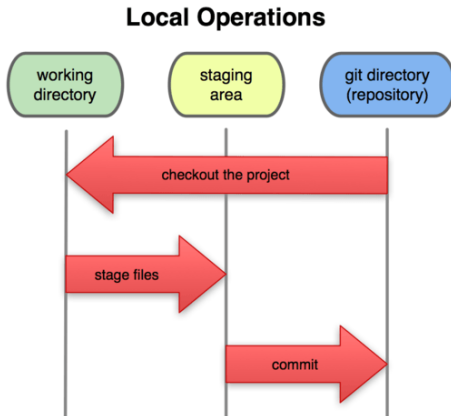


FIG. : Le titre du haut est déjà nice <3

Les bases, en images, 7/7

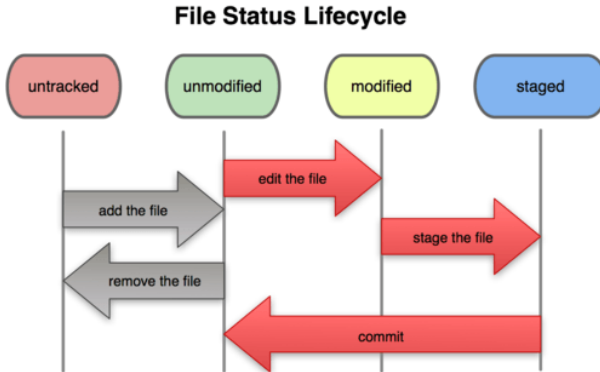


FIG. : Le titre du haut est déjà nice <3

Les principales commandes

- git init pour transformer un dossier en dépôt git
- git add pour tracker des fichiers ou dire qu'on veut les commit
- git commit pour commit (j'espère que mon explication sera meilleure à l'oral)
- git push pour mettre les derniers commits sur un dépôt lointain
- git pull pour récupérer les derniers commits sur un dépôt lointain
- git log pour avoir l'historique d'un dépôt
- git diff pour connaître les différences entre un fichier à des moments 1 & 2

Demo time





Kezako ?

Logiciels qui servent à :

- construire des programmes C/C++ (principalement)
- gérer les différents systèmes cibles
- simplifier la compilation et la re-compilation

La base : le Makefile

Fichier qui sert à exprimer les dépendences de compilation et les manières de compiler pour automatiser le build.

Demo time



La base en mieux : Makefile.in et configure

On rend le Makefile configurable en y laissant des points de customisation et on se sert de configure pour configurer ce patron de Makefile.

Demo time



La rolls royce : automake

On peut écrire un Makefile.in automatiquement. C'est ce que fait automake.

Demo time



maven

Kezako ?

Pareil qu'Autotools, mais pour la JVM, et en mieux.

Pour apprendre

<https://maven.apache.org/guides/>

Buts de Maven

Aider dans ces différentes tâches (prises du Getting Started) :

- Builds
- Documentation
- Reporting
- Dependencies
- SCMs
- Releases
- Distribution

Demo time



Mais ce n'est pas tout

- tests
 - intégration continue
 - codebase health monitoring
- bug tracker
- documentation
- ...

Demo time



Ce n'est qu'un au revoir

