

# Git

Guide pratique de démarrage (presque) en douceur

soon



Guillaume Moigneu | [guillaume@agence-soon.fr](mailto:guillaume@agence-soon.fr) | v1.0

# Planning

- 1 Rapide introduction
- 2 Quel intérêt ?
- 3 Workflows types
- 4 Processus SOON
- 5 Questions ?

prononciation : [gít]

“guite”

système de gestion du *Kernel Linux*

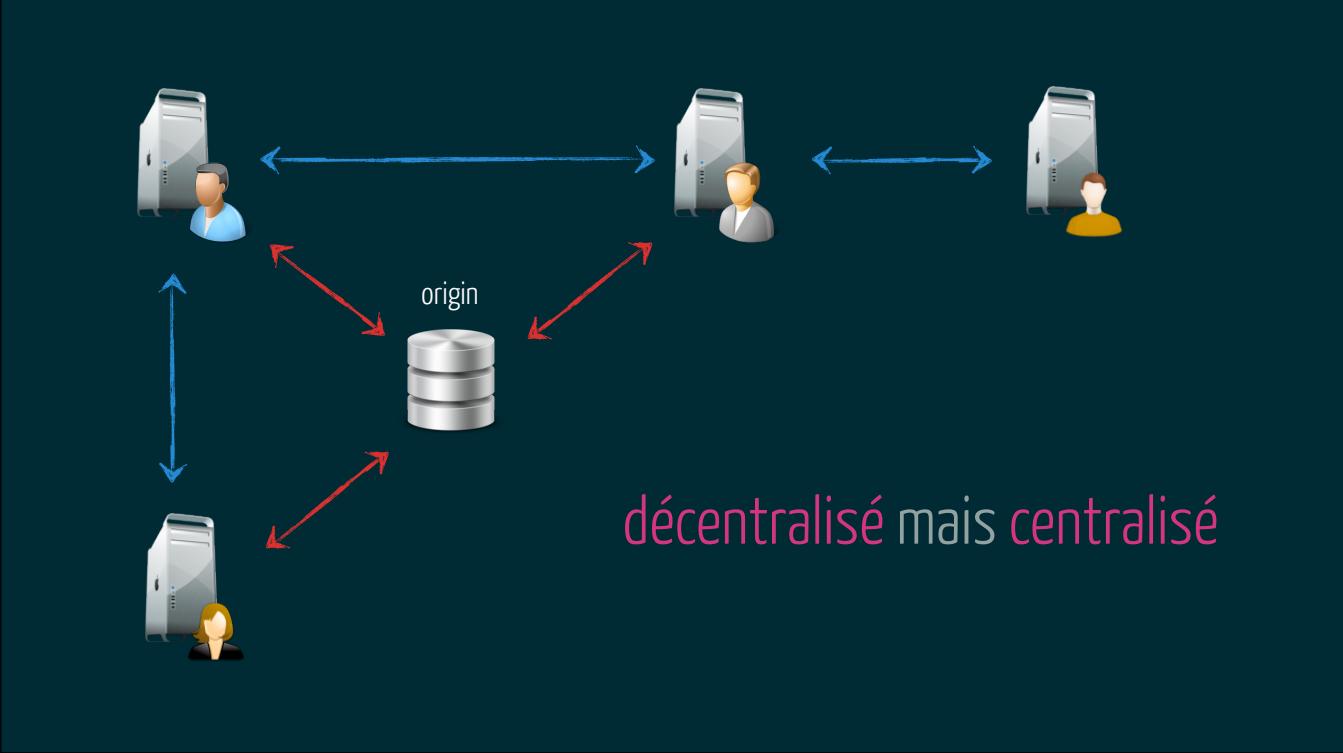
créé par Linus Torvalds



Des utilisateurs de **SVN** ?

master != trunk

branch, branch, branch.  
merge, merge, merge. tag.



git is **forgiving**

git is **fast**

git **works** on your OS

## Pourquoi Git ?

git is **easy**

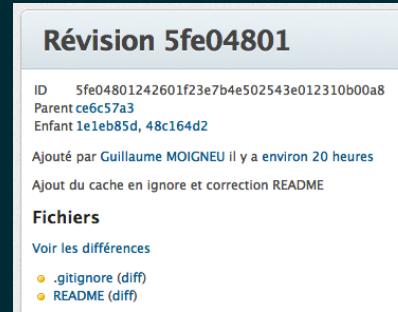
git is **geekier** (aka CLI)

git ❤️ branches

<http://whygitisbetterthanx.com>

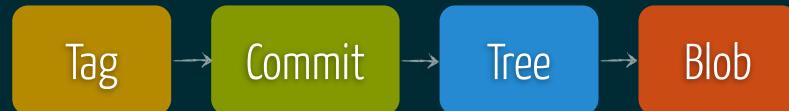
# commit ?

- lot de modifications
- identifié par un hashtag unique
- référence d'un arbre de blob
- relations entre les commits



tag?

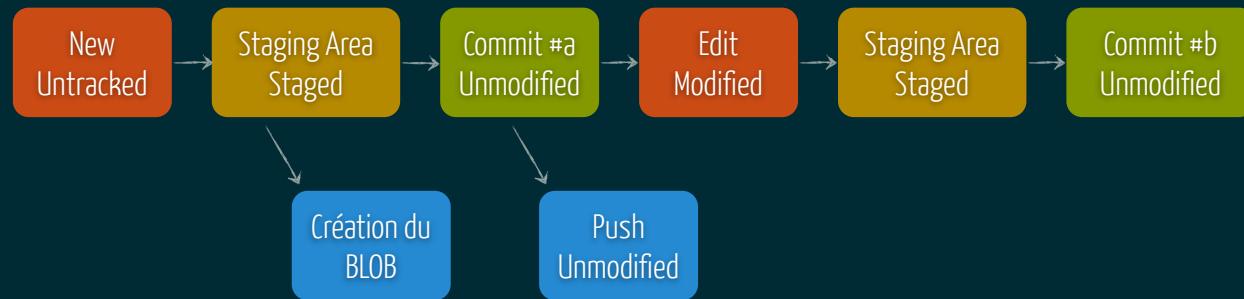
alias de commit



## Workflow de base Git

```
$ git clone git@git.agence-soon.fr:/repo-test.git  
$ vim README  
$ git add .  
$ git commit -m "Ajout du README"  
$ git push  
  
$ vim README  
$ git commit -m "Modification du README"  
$ git push
```

# Cycle de vie d'un fichier



# Diff

git status

git diff (Compares working directory with staging area)

git diff --staged

git diff HEAD

git diff v1.0 v1.1

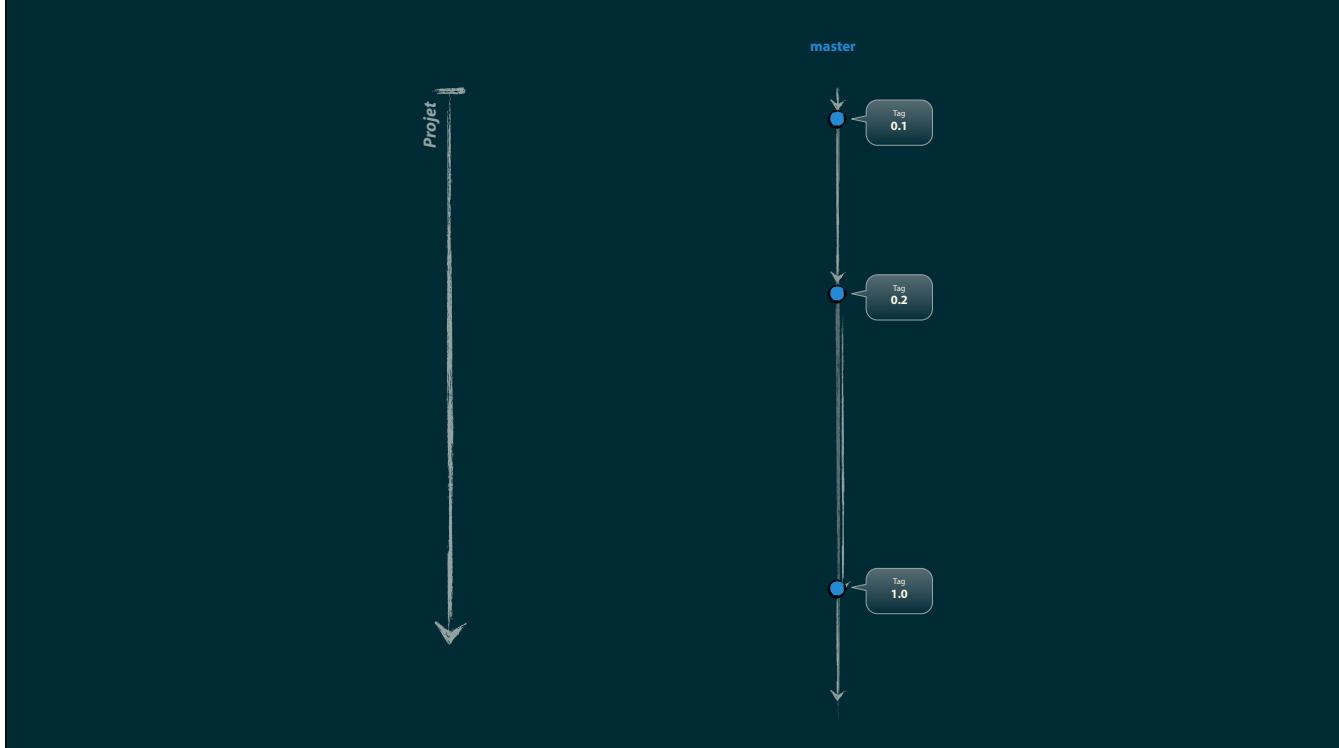
git diff master feature-1

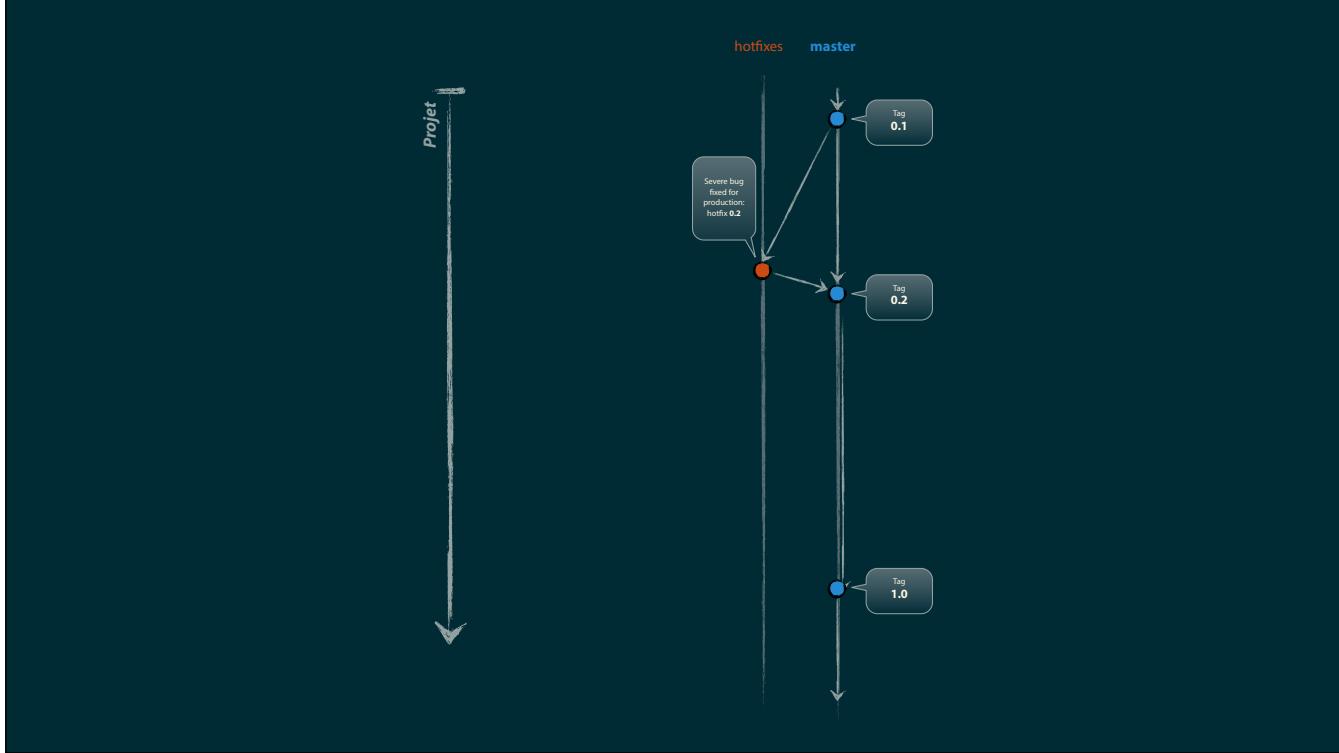
# .gitignore

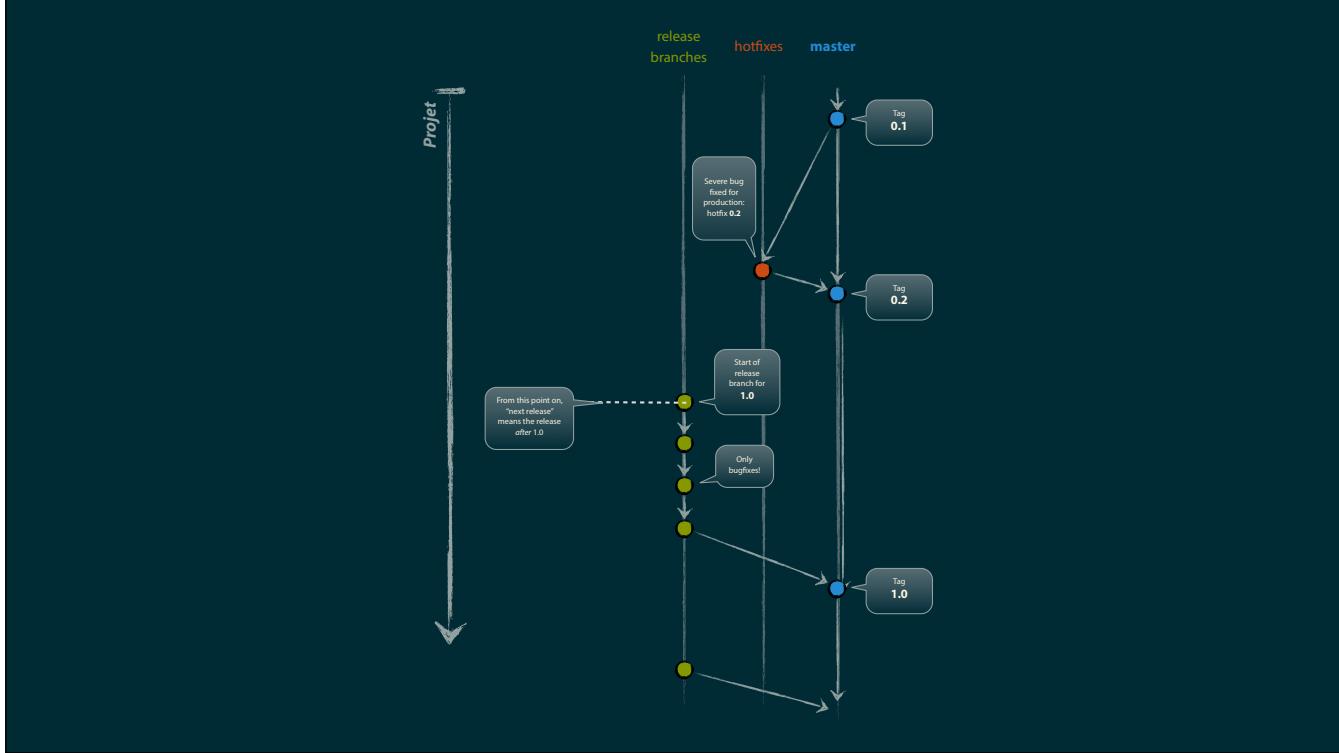
```
niels@idunn ~/Documents/Projets/en-bon-pere-de-famille [master *]
± % cat .gitignore
local-config.php
.DS_Store
sitemap.xml.gz
sitemap.xml
content/themes/bpf/sass-cache/*
```

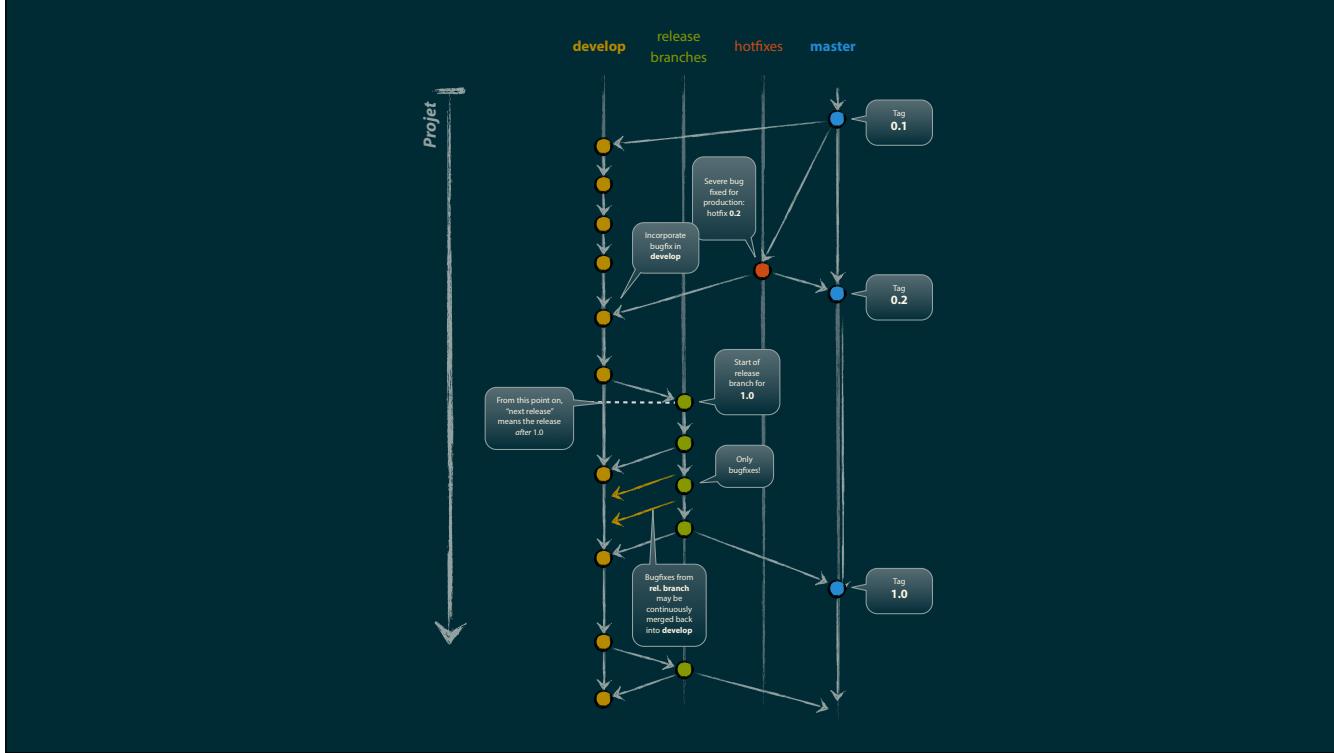
# Modèle de branches

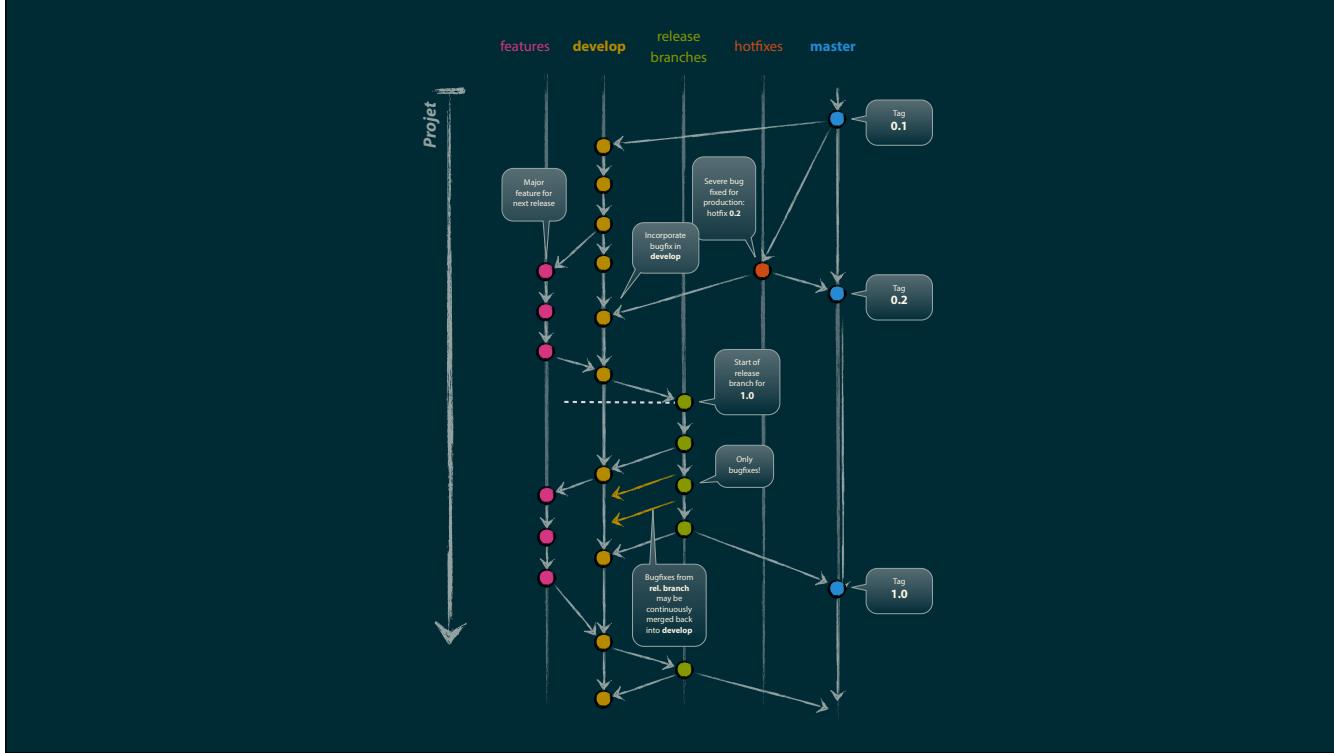
• Modèle de branches

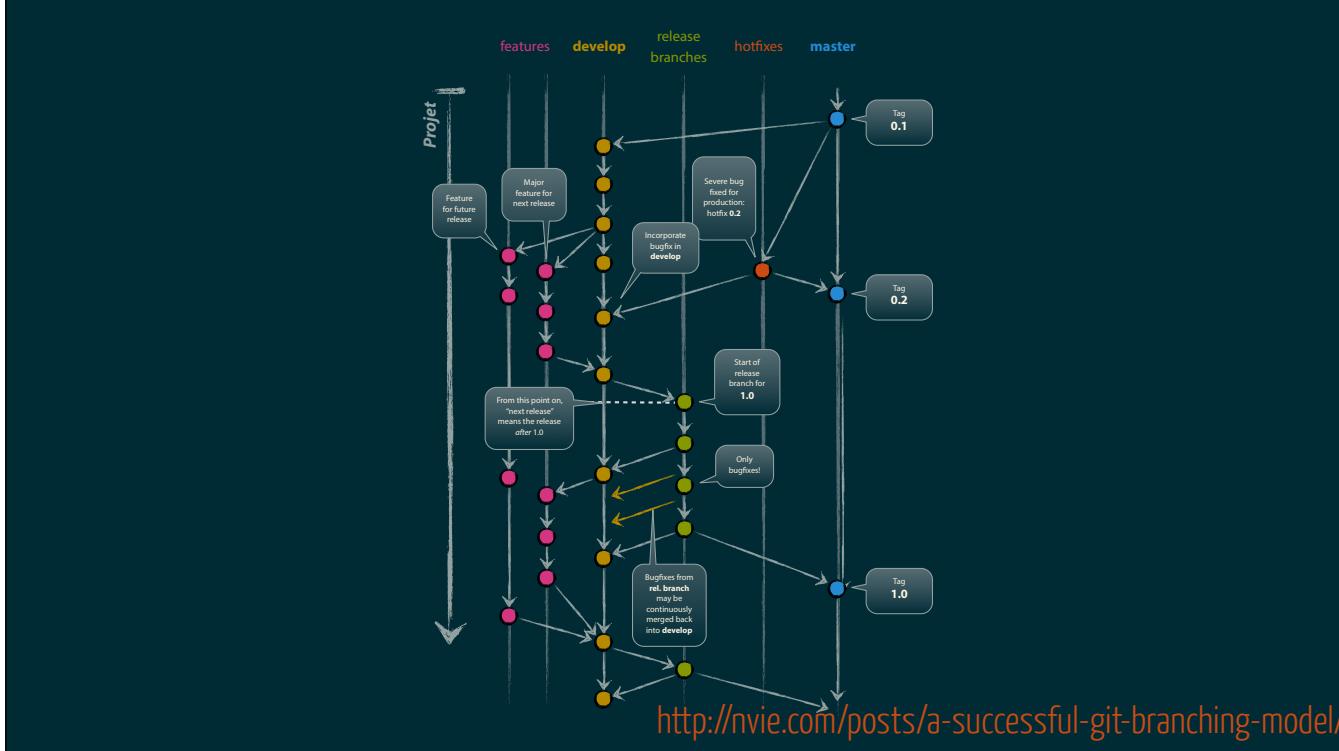




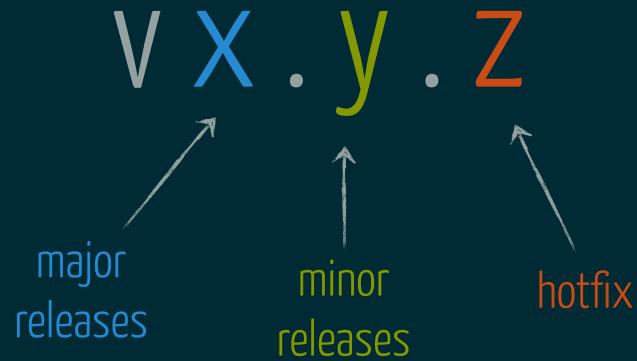








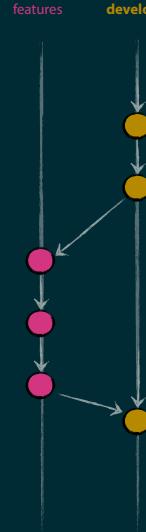
# Numérotter les versions



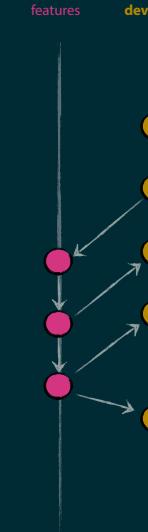
# Votre Workflow de base Git

```
$ git clone git@git.agence-soon.fr:/repo-test.git
(master) $ git checkout develop
(develop) $ git checkout -b task-666
(task-666) $ vim README
(task-666) $ git add .
(task-666) $ git commit -m "Ajout du README"
(task-666) $ git checkout develop
(develop) $ git merge --no-ff task-666
```

merge, oui,  
mais en douceur :



git merge



git rebase



# Conflits ?

```
(develop) $ git commit  
Auto-merged file.txt  
CONFLICT (content): Merge conflict in file.txt  
Automatic merge failed; fix conflicts and then commit the result.
```

```
<<<<< HEAD:file.txt  
Hello world  
=====  
Goodbye  
>>>>> 77976da35a11db4580b80ae27e8d65caf5208086:file.txt
```

```
(develop) $ git add file.txt  
(develop) $ git commit
```

# merge gone wrong ?

```
# Pas encore commit  
(develop) $ git reset --hard HEAD
```

```
# Déjà commit  
(develop) $ git reset --hard ORIG_HEAD
```

# Setup

- <http://help.github.com/set-up-git-redirect>
- <http://git-scm.com/download>

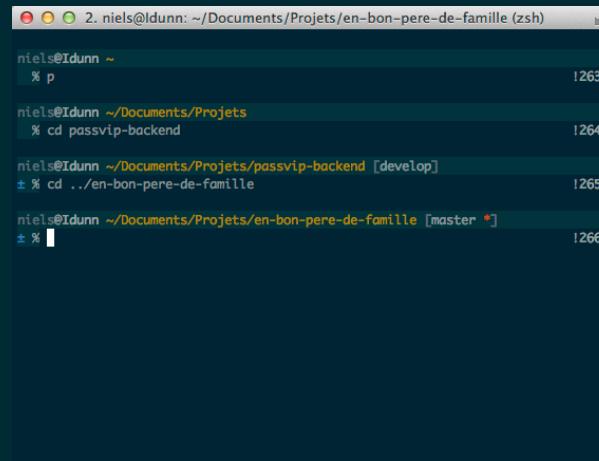
# Configurer votre shell Debian

```
aptitude install git-core git-completion zsh
```

```
chsh => /bin/zsh
```

```
wget --no-check-certificate https://github.com/robbyrussell/oh-my-zsh/raw/master/tools/install.sh -O - | sh
```

```
vim .zshrc => ZSH_THEME="blinks"
```



```
niels@Idunn ~
% p
niels@Idunn ~/Documents/Projets
% cd passvip-backend
niels@Idunn ~/Documents/Projets/passvip-backend [develop]
± % cd ../en-bon-pere-de-famille
niels@Idunn ~/Documents/Projets/en-bon-pere-de-famille [master *]
± %
```

# Outils

- <http://code.google.com/p/gitextensions/>
- <http://book.git-scm.com/>
- Desktop clients : Tower (Mac), SmartGit (Windows & Linux)
- (Bonus) git clone <https://github.com/Marak/The-Git-Rap.git>

# Notre workflow

- 1 Centralisation des repos sur vmtest
- 2 Développement en local sur vos machines
- 3 Tests sur VMs à dispositions
- 4 Recette client centralisée
- 5 Automatisation des déploiements

## Démo & exemples

**== Clone ==**

```
git clone git@git.agence-soon.fr:/projet-test.git  
cd projet-test/  
ls  
git status
```

**== Commit ==**

```
git checkout develop  
git checkout -b feature-1  
mate .gitignore  
git add .  
git commit -m "#1 feature-1 : Thème SASS"
```

**== Merge ==**

```
git checkout develop  
git merge feature-1  
git branch -d feature-1  
git push
```

**== Test ==**

```
git clone git@git.agence-soon.fr:/projet-test.git  
cd projet-test/  
ls  
git log  
git tag stable-1 id_commit
```

**== Deploy ==**

```
cd /home/httpd/  
git clone git@git.agence-soon.fr:/projet-test.git projet-  
test.vmtest.agence-soon.fr
```

**== Hotfix ==**

```
git checkout -b feature2  
vim index.html  
git checkout master  
git pull  
git checkout -b hotfix1  
vim index.html  
git add .  
git commit -m "#2 hotfix1"  
git merge hotfix1  
git branch -d hotfix1
```

**== Deploy Hotfix ==**

```
git pull
```

**== Revert ==**

```
git checkout id_commit
```

# Compléments

```
# Activer la compression  
(develop) $ git gc
```

```
# Sous-modules (a.k.a SVN External)  
(develop) $ git submodule add git@github.com:myPlugin lib/myPlugin
```

Q & A?

Fork me on GitHub.



<git@github.com:gmoigneu/git-keynote-presentation.git>

Thanks !