

# Why aren't you using Git-flow?

**Because I wait for this  
dev meeting**

# SVN

# SVN

- Organizzato in 3 cartelle

# SVN

- Organizzato in 3 cartelle
  - trunk

# SVN

- Organizzato in 3 cartelle
  - trunk
  - branches

# SVN

- Organizzato in 3 cartelle
  - trunk
  - branches
  - tag

# SVN

- Organizzato in 3 cartelle
  - trunk
  - branches
  - tag
- l'operazione di merge è macchinosa



# SVN

- Organizzato in 3 cartelle
  - trunk
  - branches
  - tag
- l'operazione di merge è macchinosa
- tutti i commit su *trunk*

# SVN

- Organizzato in 3 cartelle
  - trunk
  - branches
  - tag
- l'operazione di merge è macchinosa
- tutti i commit su *trunk*
- tag per identificare le versioni in produzione

**Ma se abbiamo un bug  
in produzione?**

# La situazione potrebbe diventare difficile

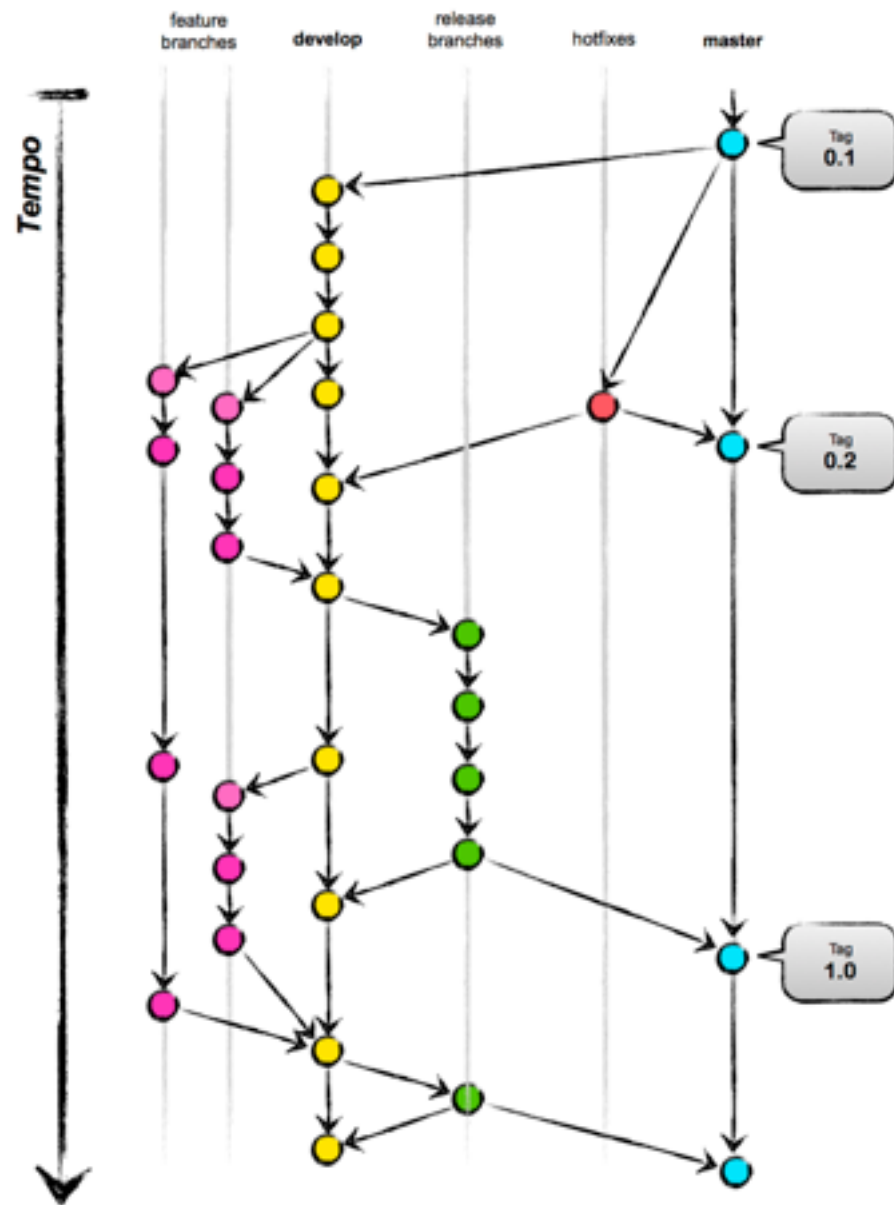
# La situazione potrebbe diventare difficile



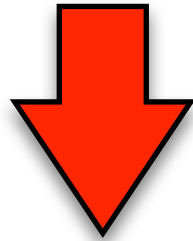
*C'è un bug in  
produzioneeeee*

# (Git) Flow

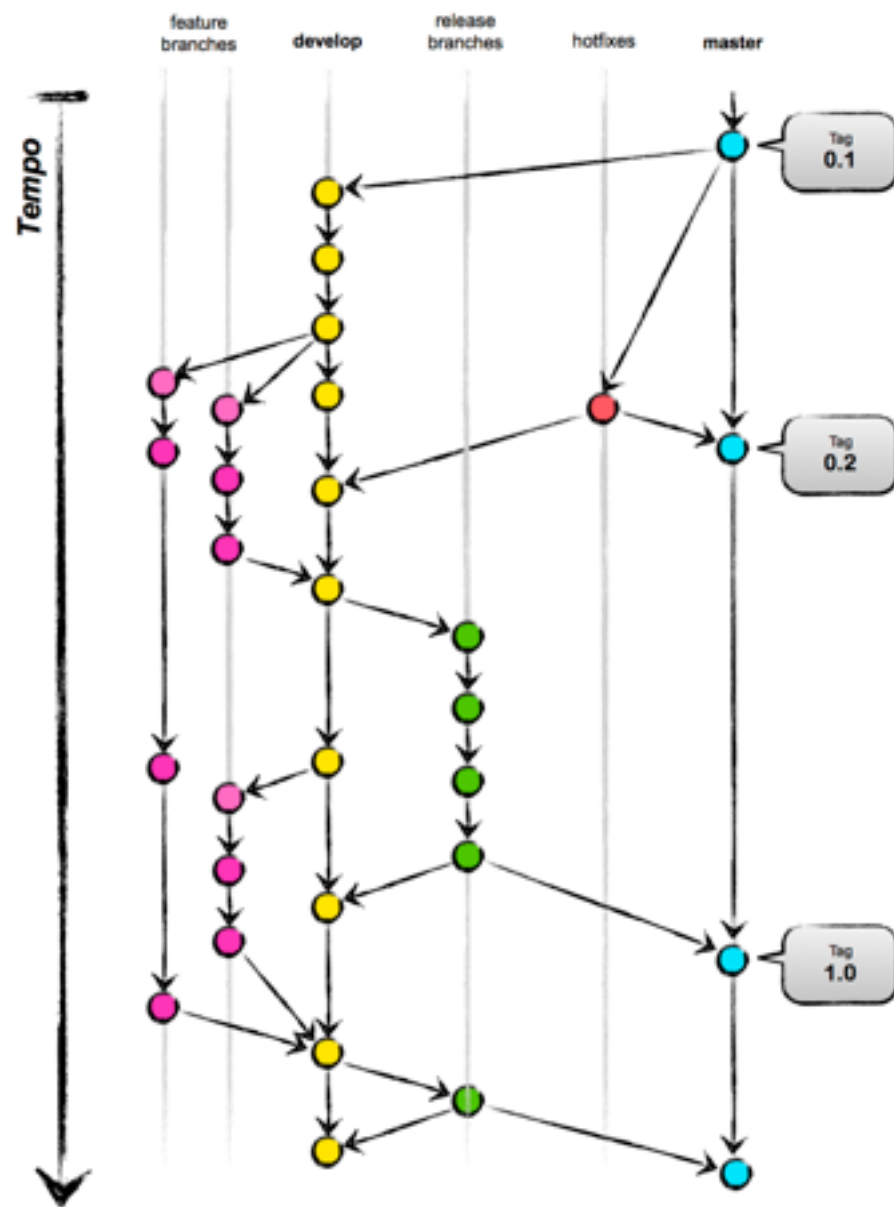
- branch principali



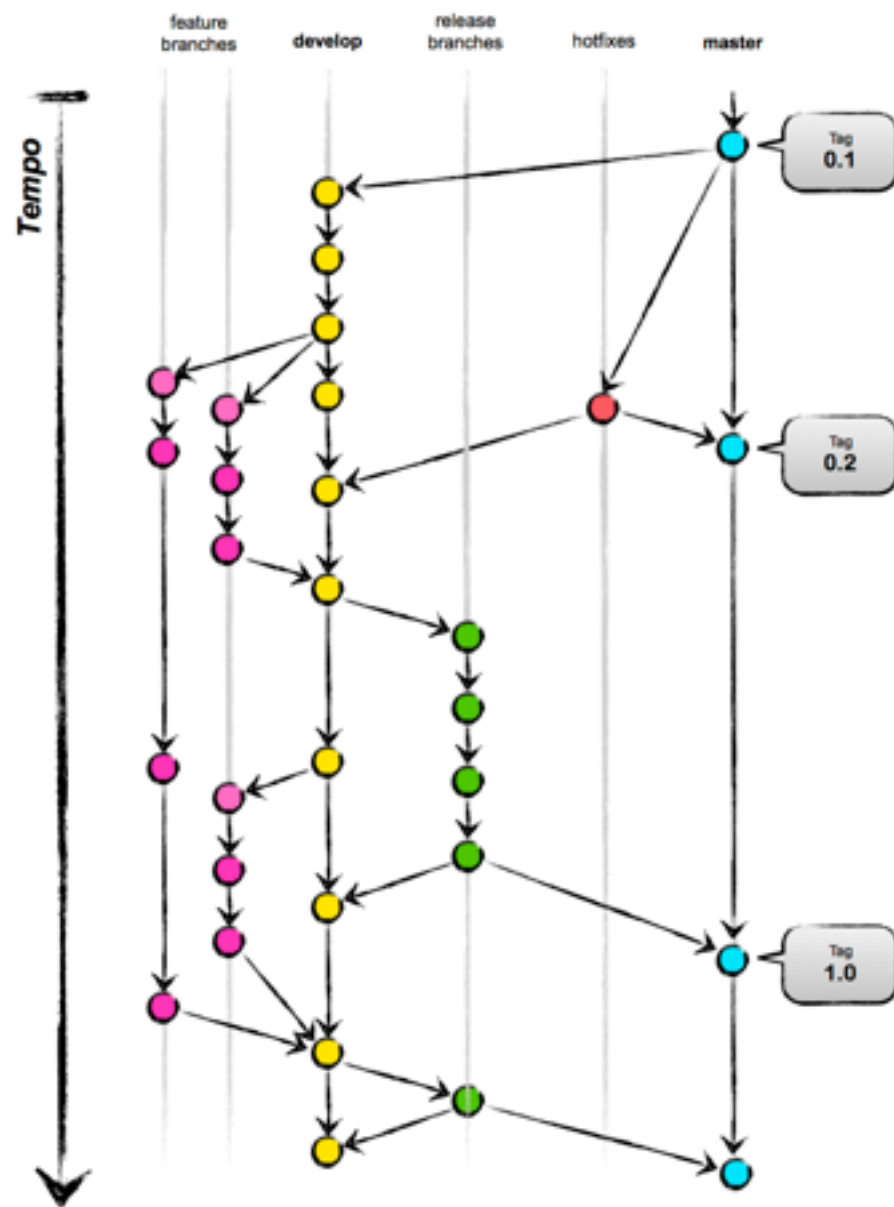
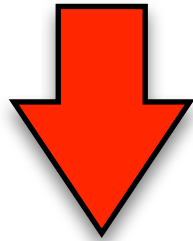
# (Git) Flow



- branch principali
- master



# (Git) Flow

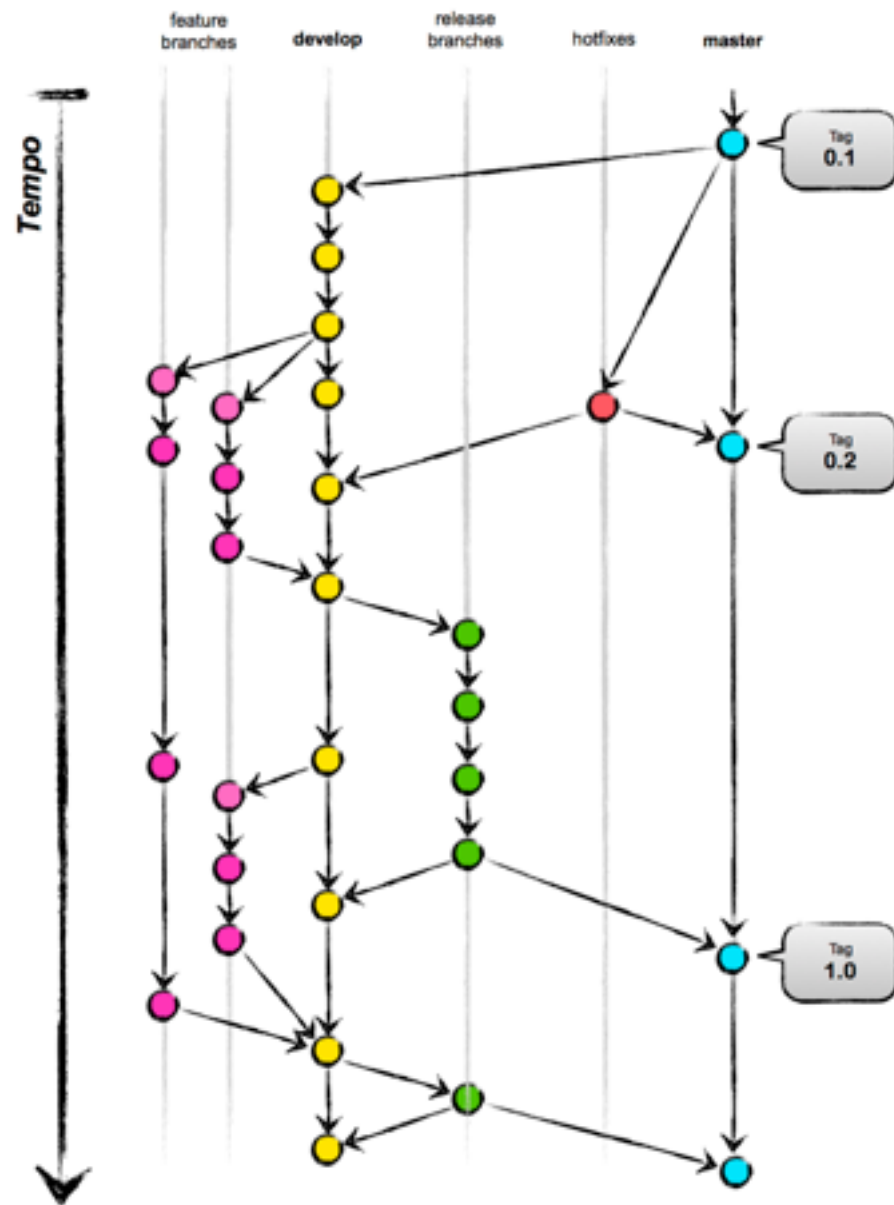


- branch principali
  - master
  - develop

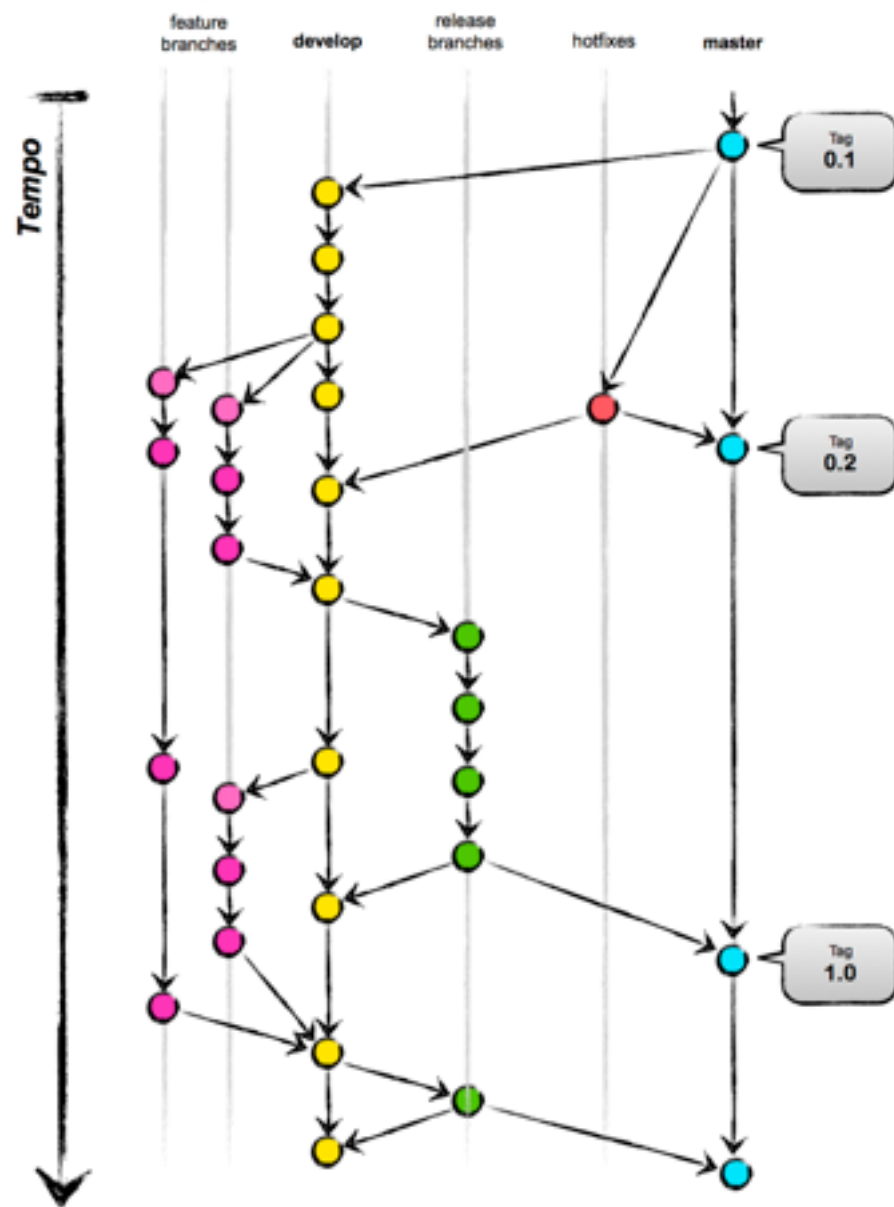
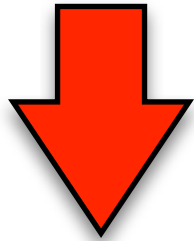


# (Git) Flow

- branch principali
  - master
  - develop
- branch di supporto

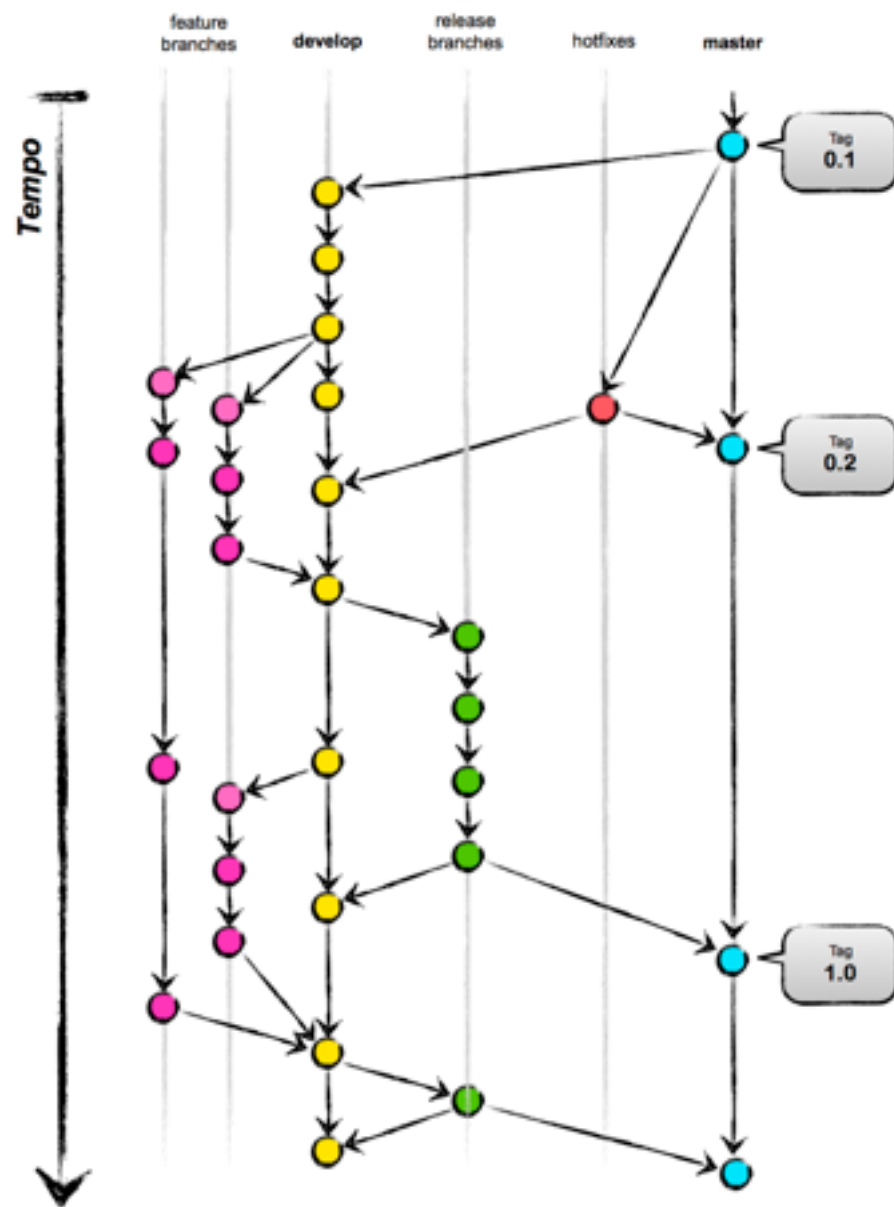
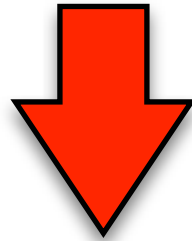


# (Git) Flow



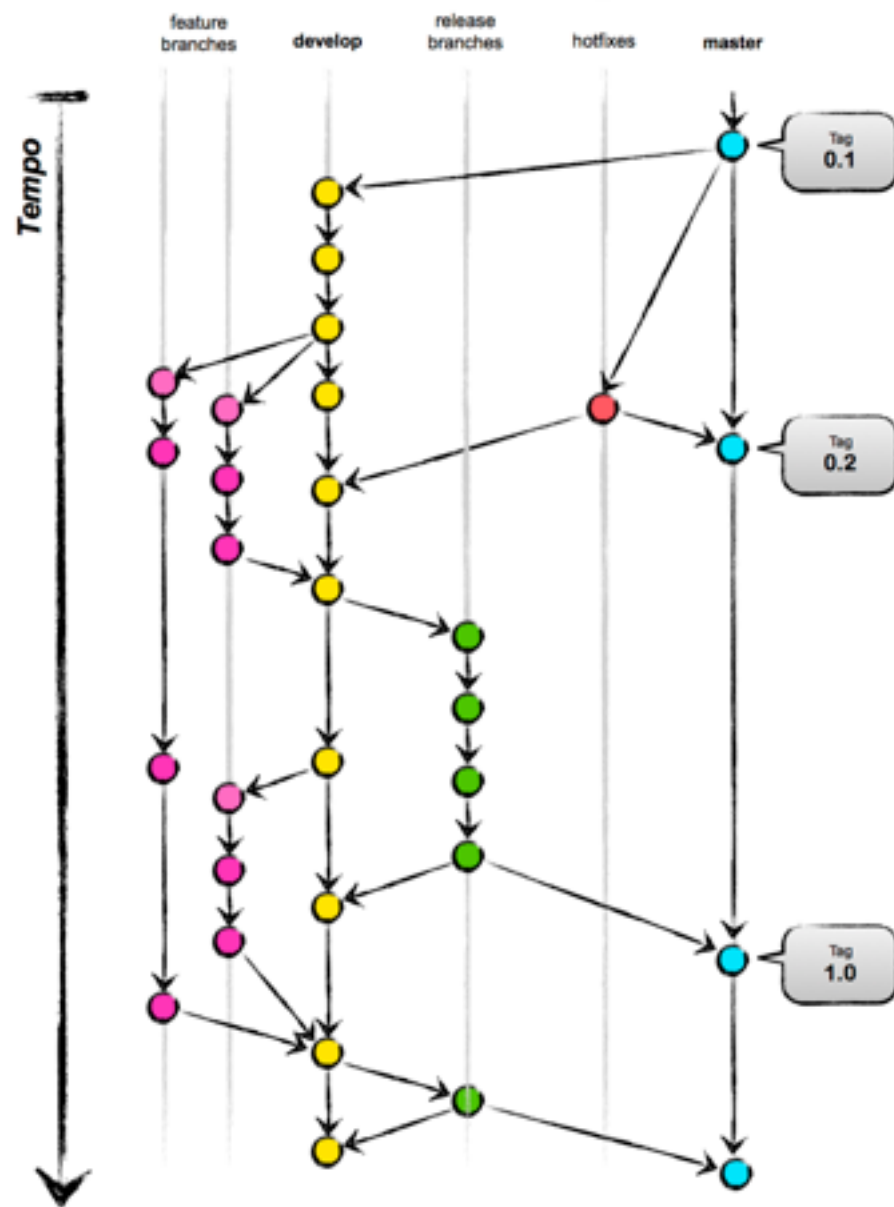
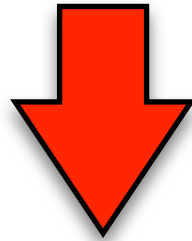
- branch principali
  - master
  - develop
- branch di supporto
  - feature

# (Git) Flow



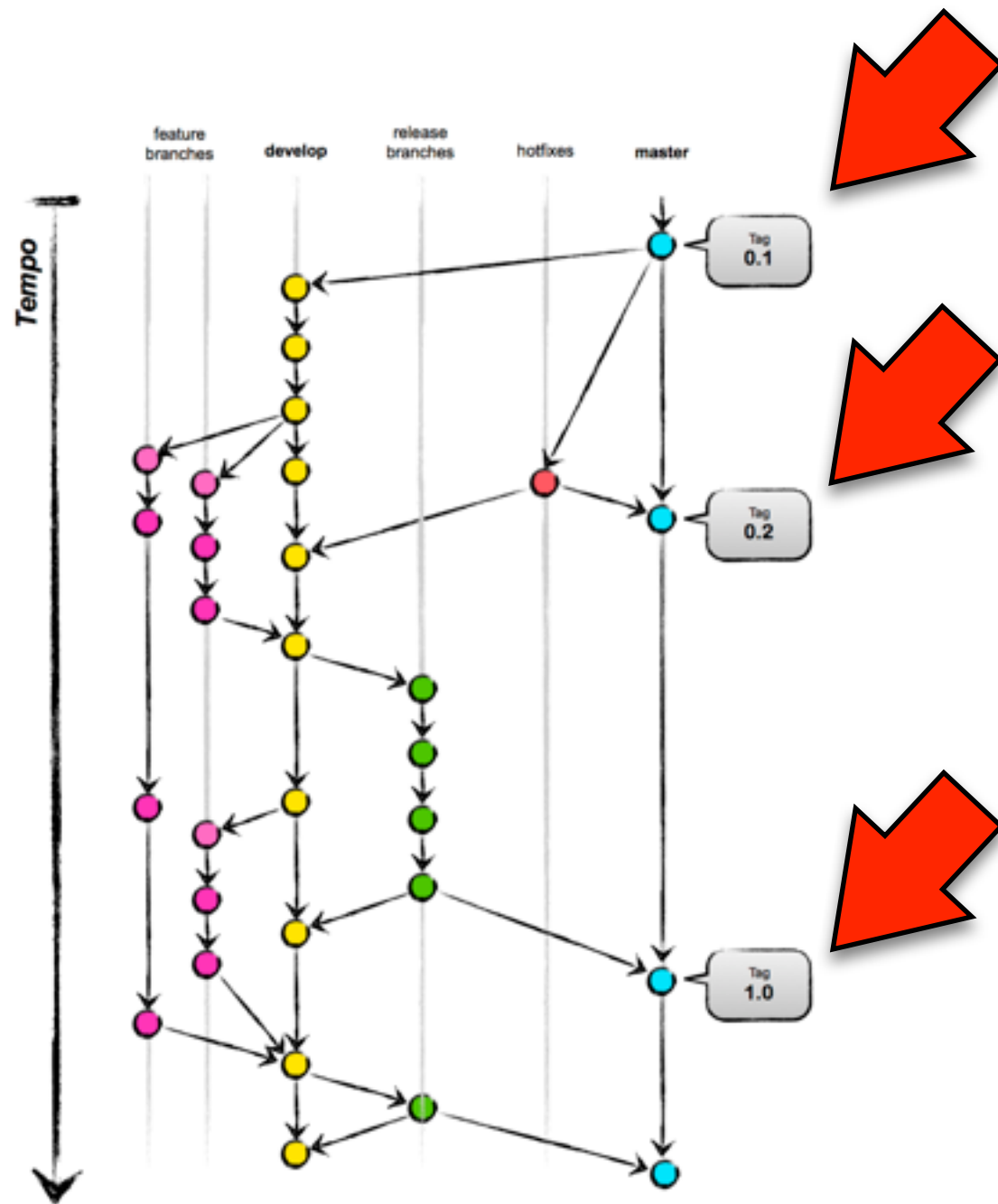
- branch principali
  - master
  - develop
- branch di supporto
  - feature
  - release

# (Git) Flow



- branch principali
  - master
  - develop
- branch di supporto
  - feature
  - release
  - hot-fix

# (Git) Flow



- branch principali

- master

- develop

- branch di supporto

- feature

- release

- hot-fix

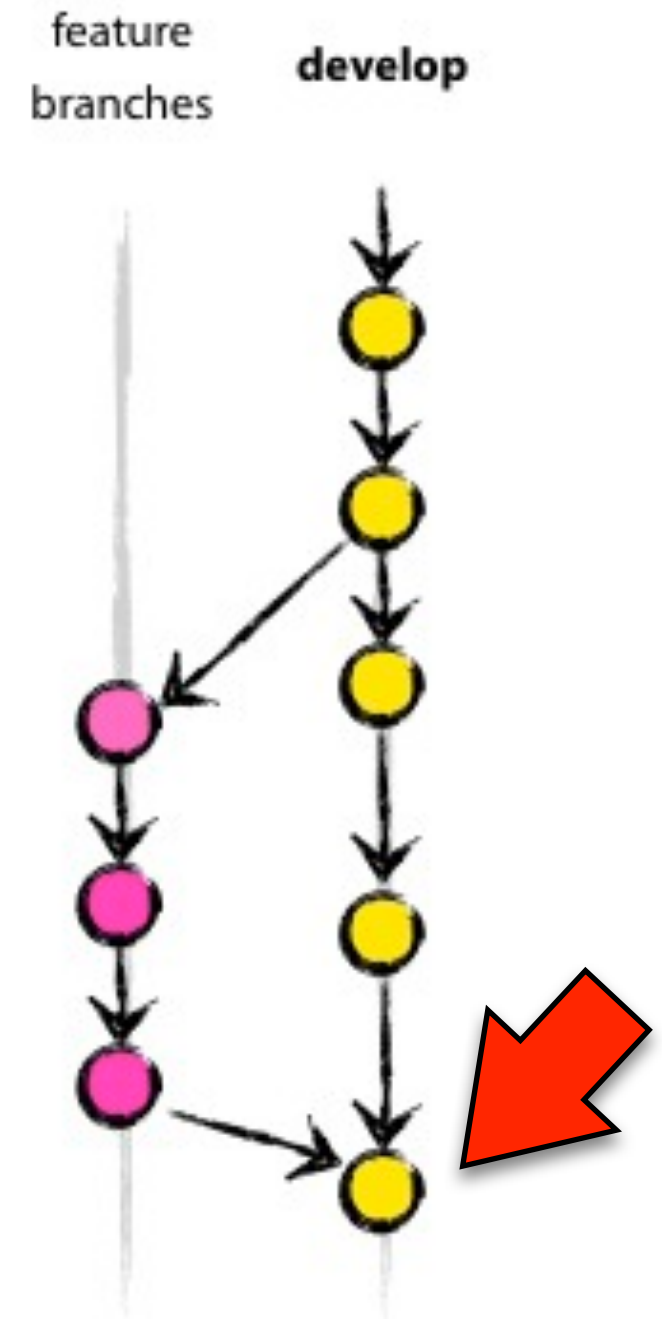
- commit su master == rilascio della release

# E se non ho mai utilizzato GIT?

- è sufficiente avere una minima conoscenza di Git
  - comandi: add, commit, branch, rebase, push, pull
  - utilizzare un front-end grafico per Git (GitX, Git Nub, Tower, Gity)
- chiedi di colleghi
- ATFG

# Feature branch

- regola
  - branch-off: develop
  - merge: develop
- funzionalità da integrare
- comandi
  - `git flow feature <name> start`
  - `git flow feature <name> finish`



# Release branch

- regola
  - branch-off: develop
  - merge: master + develop
- precede il rilascio di una release
- comandi
  - git flow release <ver> start
  - git flow release <ver> finish

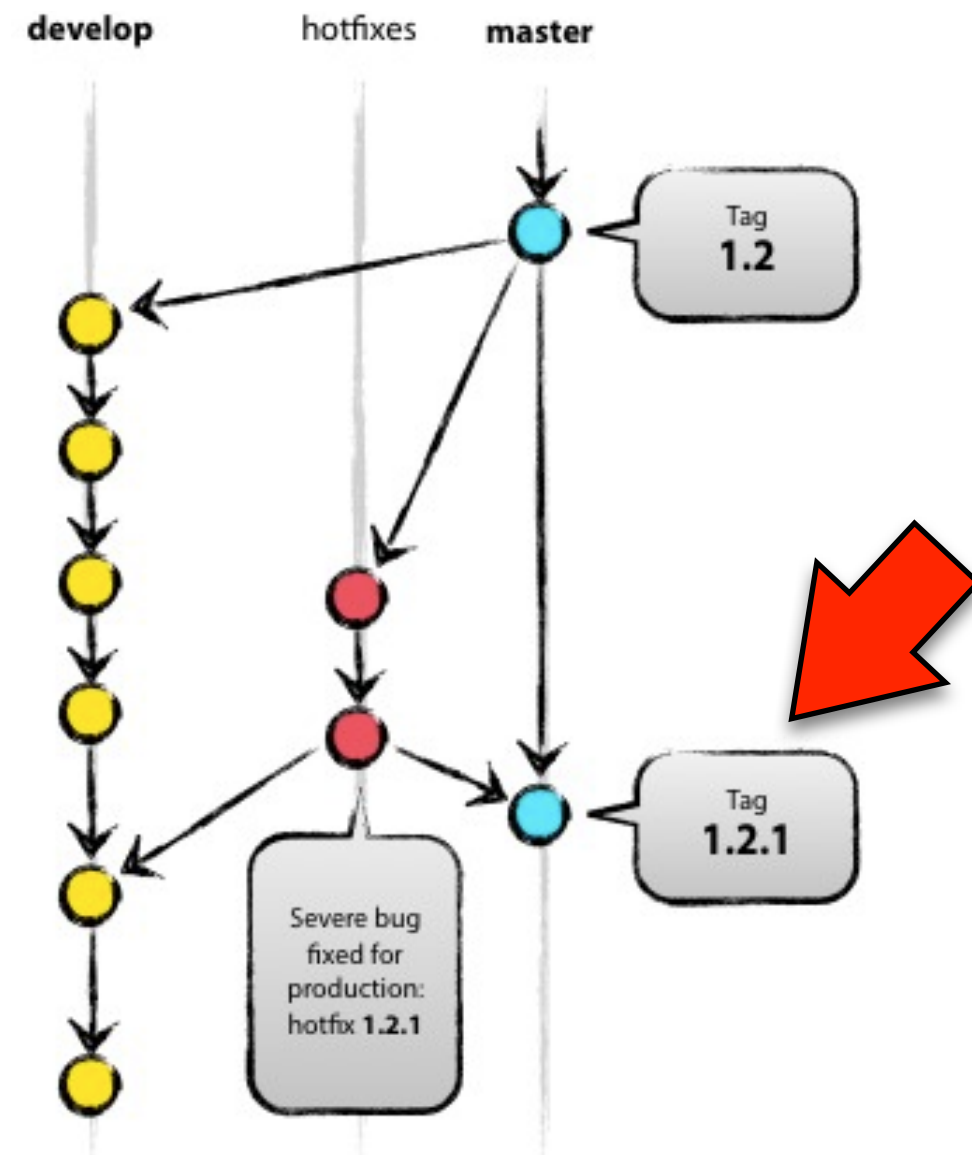




**E se abbiamo un bug in  
produzione?**

# Hot-fix branch

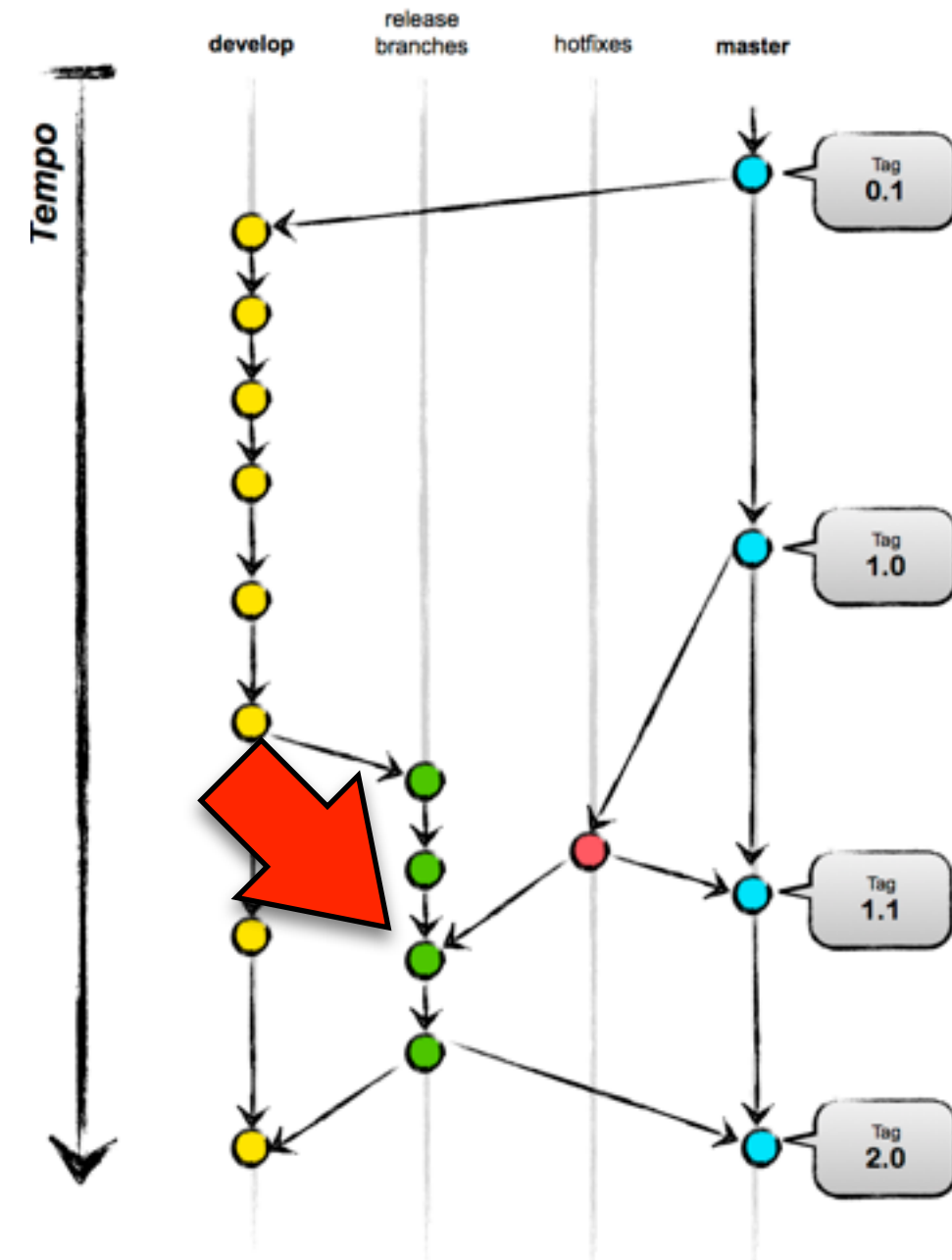
- regola
  - branch-off: master
  - merge: master + develop
- comandi
  - `git flow hotfix <ver> start`
  - `git flow hotfix <ver> finish`



**E se abbiamo un bug in  
produzione con una  
release in corso?**

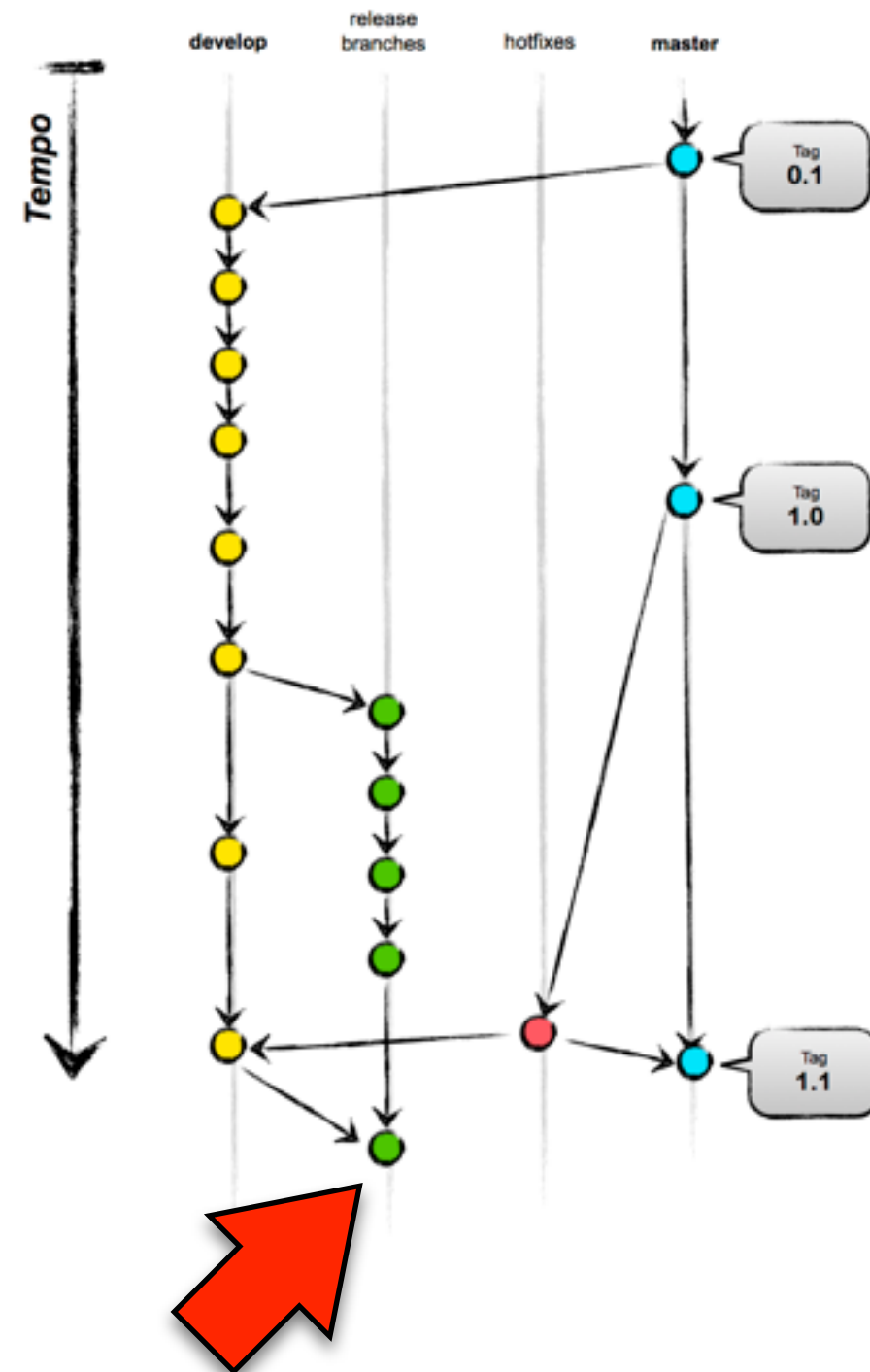
# Hot-fix branch

- regola (teoria)
- branch-off: master
- merge: master + release



# Hot-fix branch

- regola (reale)
- branch-off: master
- merge: master + develop
- release rebase develop



# Anche il bug in produzione non è più un problema



Don't try this at home

Anche il bug in produzione non è più un  
problema  
(2)



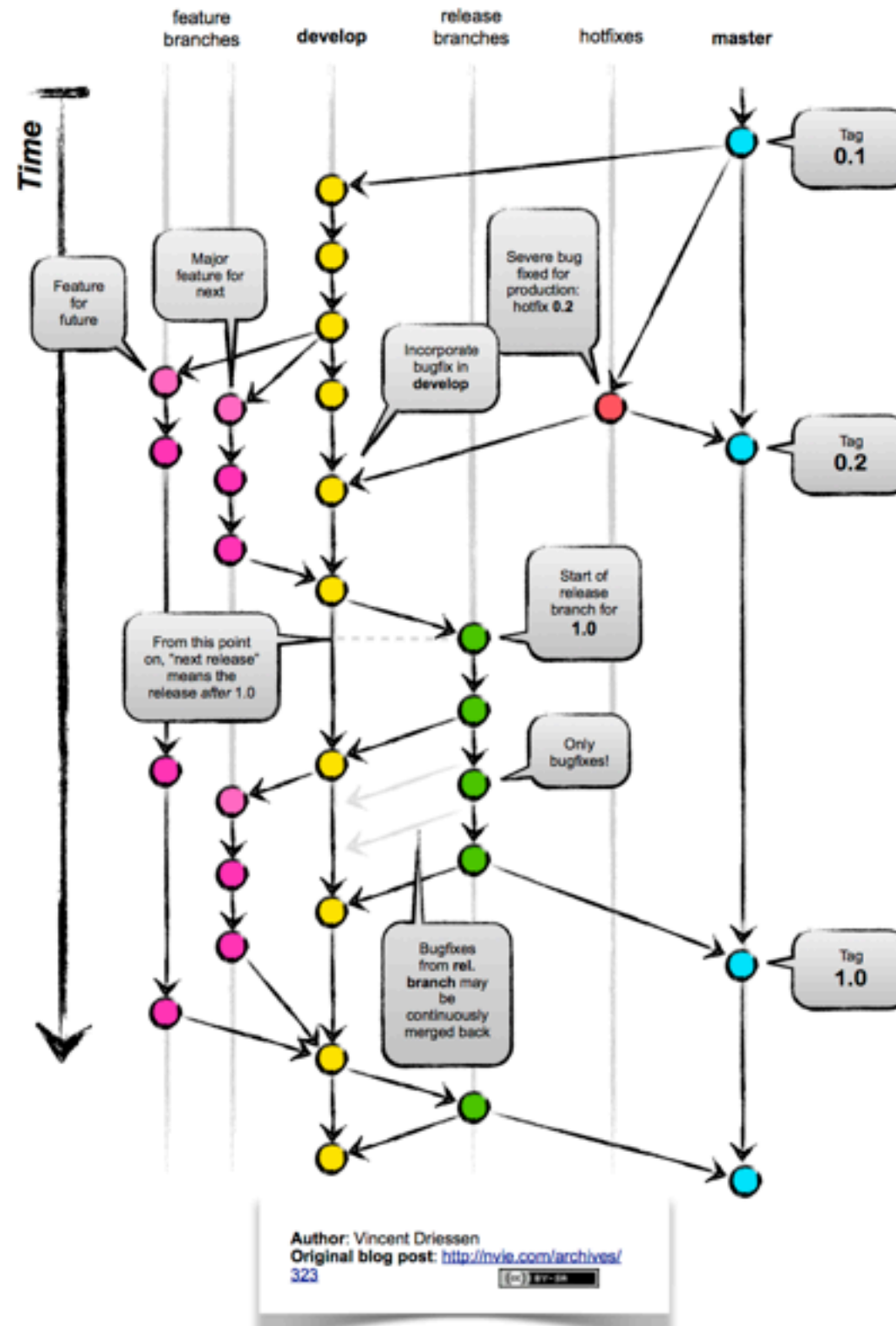
# Anche il bug in produzione non è più un problema (2)



*C'è un bug in  
produzioneeeeeee*



# Git-flow!



# Utilizzo support branch

- feature
  - git flow feature <nome> start
  - git flow feature <nome> finish
- release
  - git flow release <nome> start
  - git flow release <nome> finish
- hot-fix
  - git flow hotfix <nome> start
  - git flow hotfix <nome> finish

# Comandi principali

- `git flow init`
- *git flow feature*
  - `git flow feature list`
  - `git flow feature start`
  - `git flow feature publish`
  - `git flow feature pull`
  - `git flow feature finish`
- `git flow release`
  - `git flow release start`
  - `git flow release publish`
  - `git flow release finish`
  - `git flow hotfix`
- `hotfix`
  - `git flow hotfix start`
  - `git flow hotfix finish`

# Pro e contro

## Pro

- *struttura condivisa*
- *flessibilità*
- *semplicità*

## Contro

- *non si applica a SVN\**
- *conoscenza di Git\**
- *not H-art complaint\**
- *Git-flow crea dipendenza*

# Sitografia

- la guida: <http://nvie.com/posts/a-successful-git-branching-model/>
- guida estesa: <http://yakiloo.com/getting-started-git-flow/>
- github: <https://github.com/nvie/gitflow>
- A short introduction to Git Flow: <http://vimeo.com/16018419>
- [http://www.codesherpas.com/screencasts/on\\_the\\_path\\_gitflow.mov](http://www.codesherpas.com/screencasts/on_the_path_gitflow.mov)
- bonus: 8 ways to share your git repository: <http://www.jedi.be/blog/2009/05/06/8-ways-to-share-your-git-repository/>

# Grazie

Nessun attore è stato maltrattato durante  
la preparazione di queste slide.

Nessun collega è stato molestato durante la preparazione  
di questa presentazione.