UNIVERSITÀ DEGLI STUDI DI MILANO

Dept. of Computer Science

Lab 3: Git branching

Carlo Bellettini, Matteo Camilli

carlo.bellettini@unimi.it
matteo.camilli@unimi.it

Ingegneria del software a.a. 2016/17

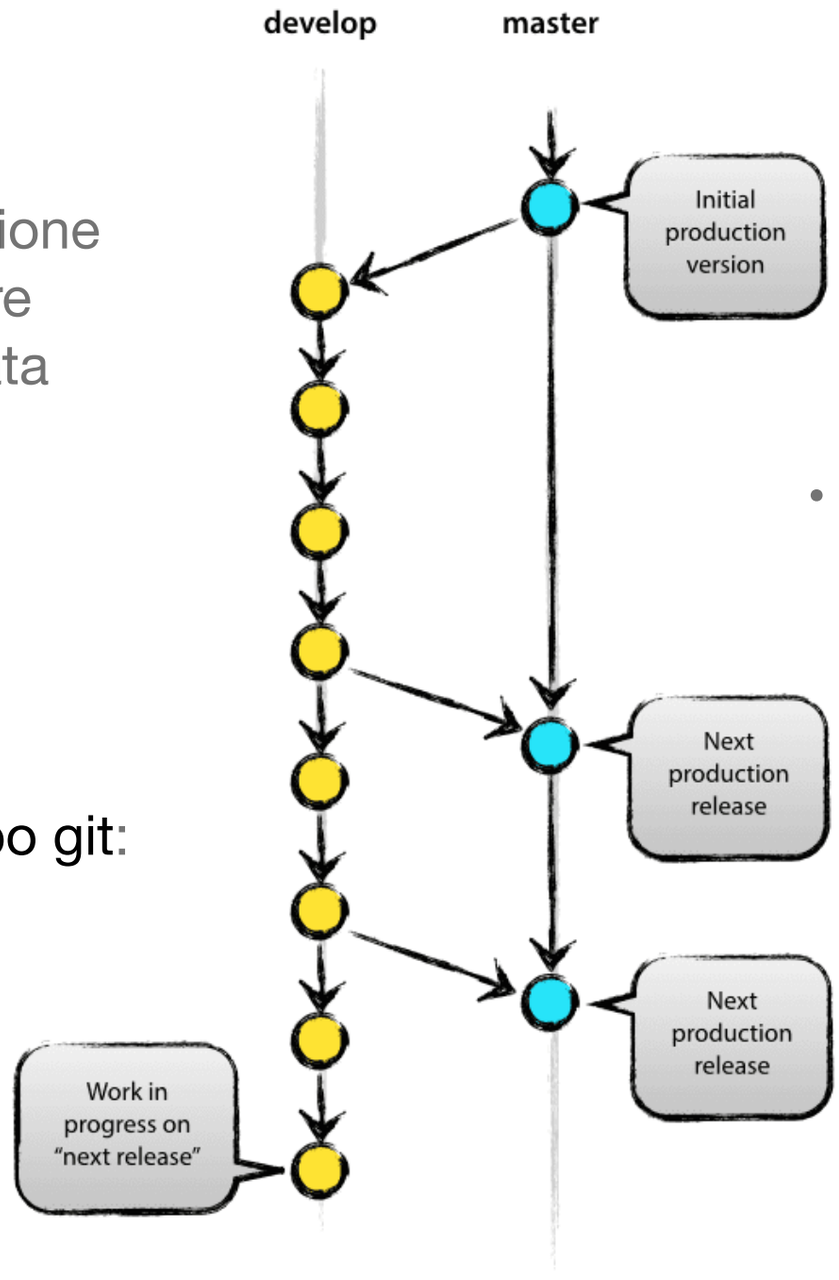
Git flow

- git-flow
  - https://danielkummer.github.io/git-flow-cheatsheet/
- installazione
  - OSX
    - Homebrew: `brew install git-flow-avh`
    - Macports: `port install git-flow-avh`
  - Linux
    - `apt-get install git-flow`

Git branching: master/develop

- develop: riflette la versione più recente del software che può essere integrata nel ramo master

- inizializzazione di un repo git:
  `git flow init`



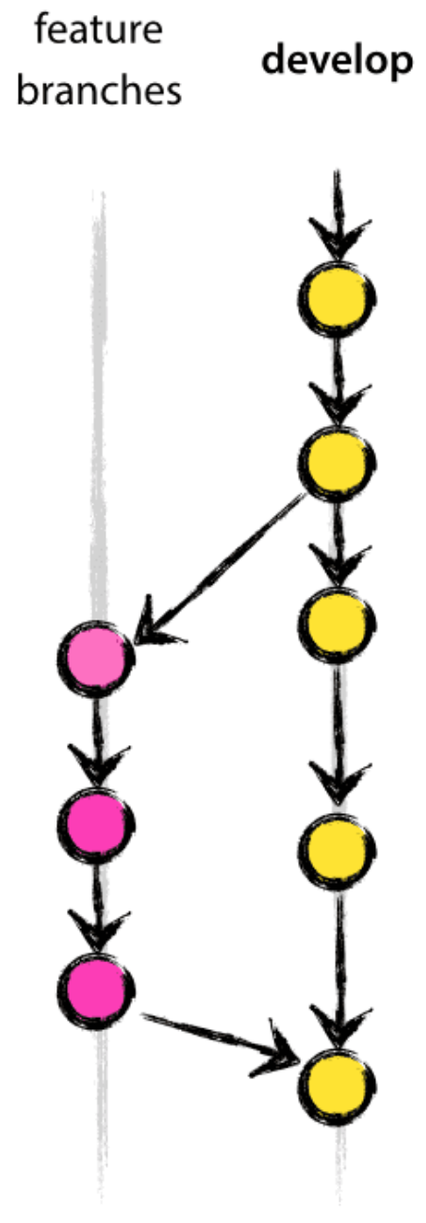
- master: rilasci del software in produzione

Git branching: feature branches

- feature branches: usati per sviluppare nuove funzionalità per versioni future del software.

  - il ramo “feature” esiste fintanto che la funzionalità è in fase di sviluppo.
  - il ramo “feature” viene infine inglobato all’interno del ramo “develop”, oppure scartato

- creare feature branches:
  `git flow feature start <nome feature>`
- effettuare il merge del feature branche in develop:
  `git flow feature finish <nome feature>`



Git branching: release

- release di una versione all'interno del ramo master
  - `git flow release start <numero versione>`
    - crea un nuovo ramo "release" a partire dal ramo "feature" e che ingloba le ultime modifiche del ramo "develop"
  - correzione di eventuali errori introdotti dal merge
  - `git flow release finish <numero versione>`
    - effettua il merge del ramo "release" nel ramo "master" e successivamente il merge del ramo "master" nel ramo "develop"

Git: sincronizzazione

- PUSH ALL
  - trasferimento di commit locali di tutti i branch da repository locale a repository remoto
    - `git push origin --all --follow-tags`

Riferimenti

- https://danielkummer.github.io/git-flow-cheatsheet/
- http://nvie.com/posts/a-successful-git-branching-model/