

SCM & Workflow

@bear

做正确的事情

VS

正确地做事情

为什么需要SCM?

- `tar zcvf 20141126.zip src && cp *.zip backup`
- `rm -rf src` = `\(rm -rf\)` WTF!

基本概念

- repo
- trunk
- branch
- revision/commit
- changeset/changelist
- tag

SVN vs GIT

- 操作单位是目录
- 中心服务器，连网
- 单repo，多分支
- 分支操作很重，耗时
- 普及率高，学习成本低
- 操作单位是文件
- 分布式，网络非必要
- 多repo，多分支
- 分支操作轻
- github/pullrequest
- 学习曲线略陡

SVN

- svn co
- svn up
- svn add
- svn mkdir
- svn ci
- svn rm
- svn mv
- svn cp
- svn sw
- svn merge
- svn revert -r dir
- svn log -l 10
- svn diff -r 1001
- svn blame bear
- svn info
- svn ls
- svn st
- svn resolved
- svn pe svn:ignore dir
- svn changelist
- svn cleanup

GIT

- git clone
- git checkout
- git add -A
- git commit
- git rm
- git mv
- git br
- git remote add
- git rebase
- git merge
- git reset HEAD
- git log --name-status
- git diff
- git blame bear
- git push
- git pull
- git fetch
- git status

stage ?
remote? rebase?

什么是Workflow?

- 编写代码->调试自测->提交->上线发布
 - 只关注源代码这个角度
 - 多人协作

几种Workflow

- trunk开发，trunk提交，trunk发布
- branch开发，merge到trunk，trunk发布
- fork repo，本地分支开发，推送到远端分支，发起pullrequest，bless分支发布

Free Style



- `svn co http://foo.com/svn/bar/trunk`
- `svn up`
- coding...
- `svn add`
- `svn ci -m "fix bug"`
 - 没有code review，或者事后review
 - 所有人都有权限

Branches



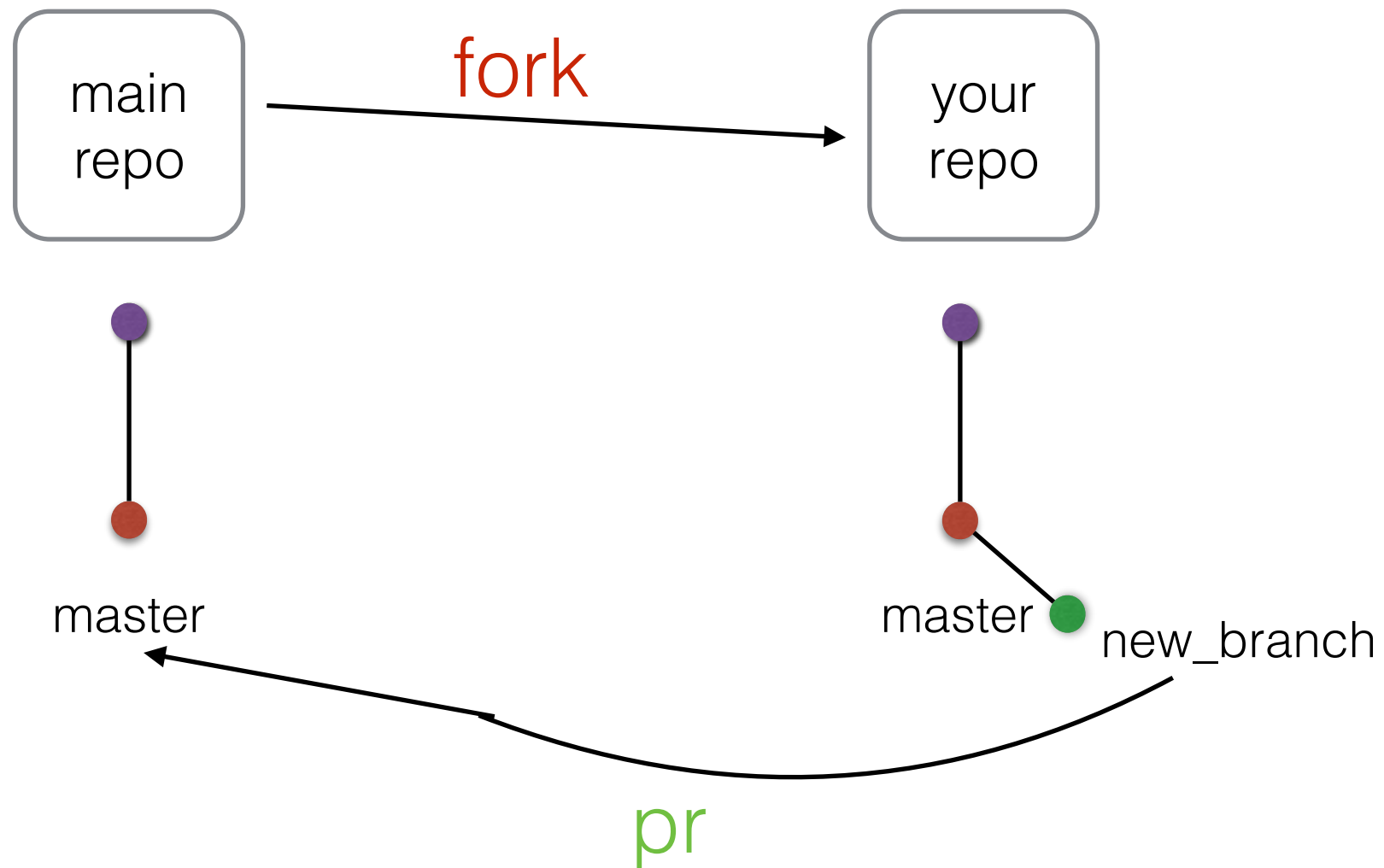
- `svn co http://foo.com/svn/bar/trunk`
- `svn cp http://foo.com/svn/bar/trunk http://foo.com/svn/bar/branches/new_branch`
- `svn sw http://foo.com/svn/bar/branches/new_branch`
- coding...
- `svn add / svn ci`
- `svn sw http://foo.com/svn/bar/trunk`
- `svn merge -r 10001:10020 http://foo.com/svn/bar/branches/new_branch`
- `svn add / svn ci`

部分人或全部人有权限，
但需要review通过后才能合并

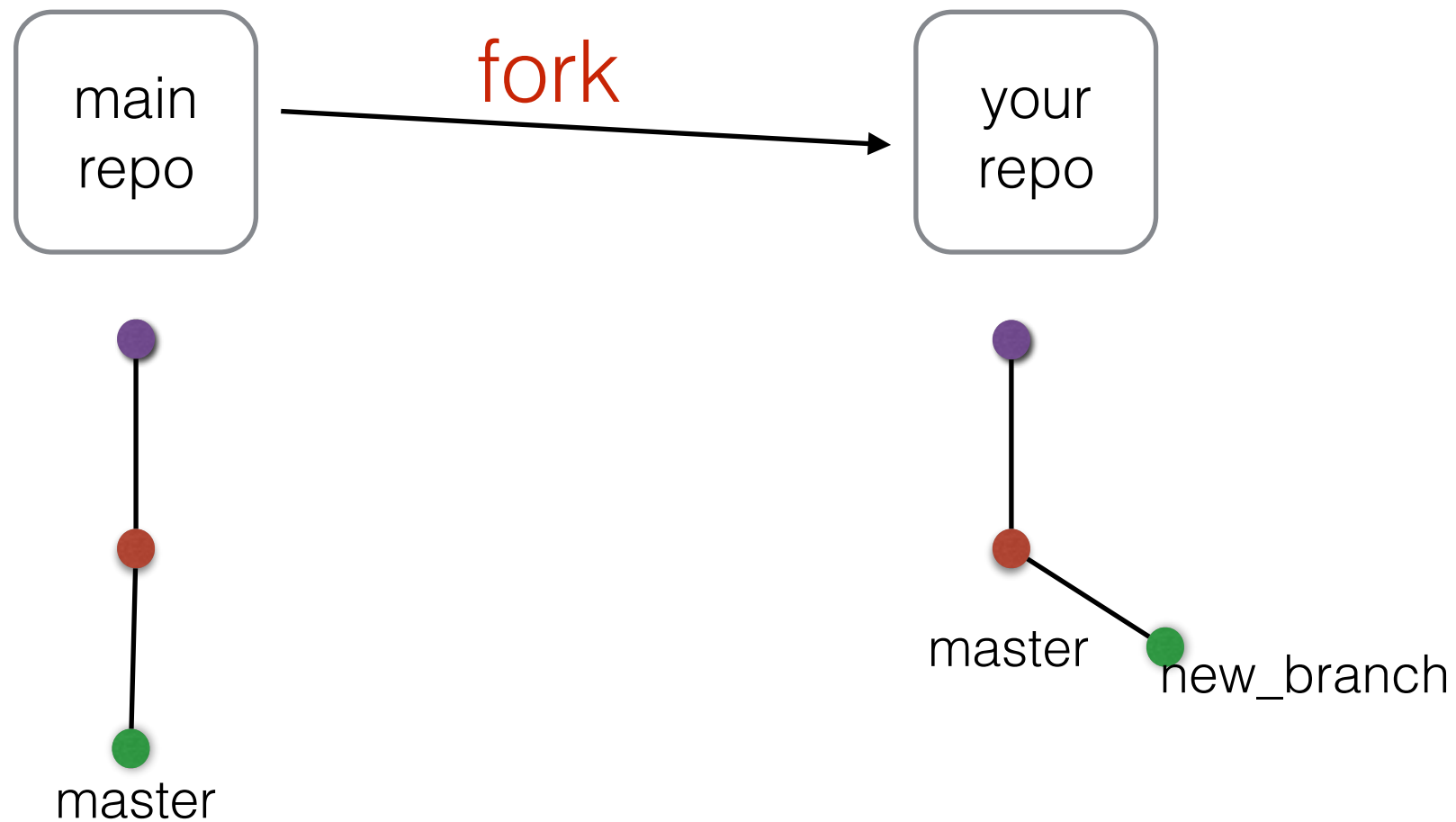
Pullrequest

- fork main_repo your_repo
- git clone your_repo.git
- git remote add upstream main_repo.git
- git branch new_branch
- git checkout new_branch
- git add -A .
- git commit -m "some"
- git push --set-upstream origin new_branch
- git push
- create your pull request (your_repo:new_branch --> main_repo:master)

Pullrequest



Pullrequest



Pullrequest

- `git fetch upstream`
- `git merge upstream/master`
- `git push`
- `git branch new_branch2`
- `git checkout new_branch2`
- coding...

没有最好的
只有最适合的