

ביקורת תצורה עם GIT & GITHUB לניהול פרויקט גמר

המכללה האקדמית להנדסה HAC

8/12/21

ראובן יגאל

<https://github.com/robi-y/presentations>

בסוף המפגש:

אמילִי – אז מה יצא לי מזה ☺

- מהי בקורת תצורה ולמה זה טוב?
- מה ההבדל בין git ל-?github
- האם זה יעזור לי בפרויקט הגמר? בתעשייה?
- ולפי הזמן, איך GIT עובד



- [HTTPS://GITHUB.COM/RAMBLINGCOOKIEMONSTER/GIT-PRESENTATION](https://github.com/RamblingCookieMonster/GIT-PRESENTATION)
- [HTTPS://GITHUB.COM/JCE-IL/SE-CLASS-MATERIALS/TREE/MASTER/LECTURE](https://github.com/JCE-IL/SE-CLASS-MATERIALS/tree/master/LECTURE)
- [HTTP://TOM.PRESTON-WERNER.COM/2009/05/19/THE-GIT-PARABLE.HTML](http://TOM.PRESTON-WERNER.COM/2009/05/19/THE-GIT-PARABLE.HTML)

איך נעשה את זה?
- מבוא לברית
- תצורה
- קצר על גיט
- וGITהאב
- תרגול
- "משל גיט"

A Crash Course in Version Control and Git

Warren Frame

Agenda



GitHub



git

What is Version Control?

- System to manage changes to files

- Who vors
- When April 17
- What Files and differential
- (Sometimes) Why To fix Pester tests

- Why use it?

- Revert to or review prior changes
- Maintain multiple versions
- Compare differences
- Share and collaborate
- Modern solutions might assume you have it!
 - Infrastructure-as-code
 - Continuous Integration and Delivery
- Google around for many more reasons!

Fix pester tests for MSFT_xADRecycleBin module

master (#1) 2.4.0.0-PSGallery

vors authored on Apr 17 1 parent 35e663e

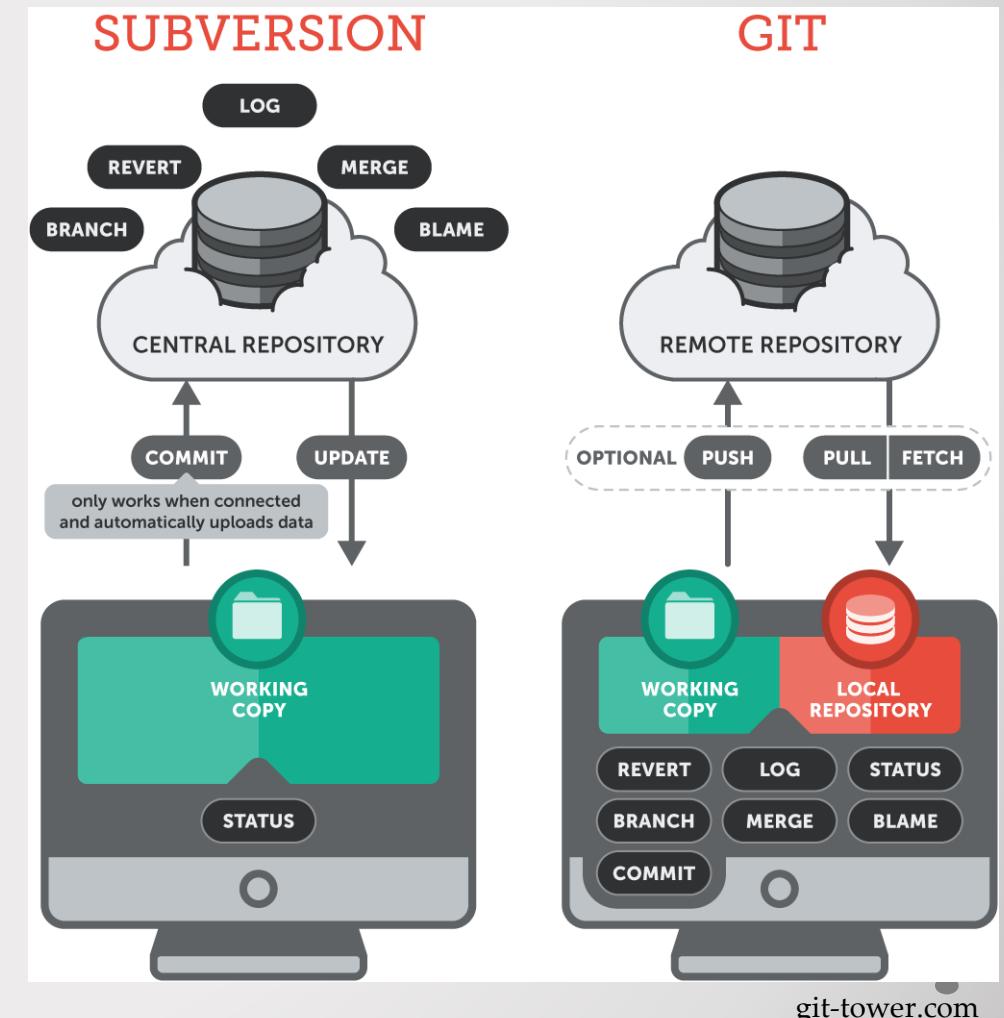
Showing 2 changed files with 16 additions and 8 deletions.

16 DSCResources/MSFT_xADRecycleBin/Tests/ResourceDesigner.Tests.ps1

| ... | ... | @@ -1,11 +1,11 @@ |
|-----|-----|---|
| 1 | 1 | + Describe 'xADRecycleBin' { |
| 2 | | - Context 'xDscResourceDesigner' { |
| 3 | | - Test-xDscResource xADRecycleBin -Verbose Should Be \$True |
| 4 | | - } |
| 5 | | -} |
| | 2 | + Context 'xDscResourceDesigner' { |
| | 3 | + It 'Pass Test-xDscResource' { |
| | 4 | + Test-xDscResource xADRecycleBin -Verbose Should Be \$True |
| | 5 | + } |

Version Control Systems

- Centralized
 - Work directly against a central server
 - Subversion, CVS, Perforce, etc.
- Distributed
 - Work locally, optionally push to remote repositories
 - Git, Mercurial (Hg), etc.
- Hosted solutions
 - GitHub, BitBucket, Visual Studio Online, etc.
- On-Premise solutions
 - GitHub Enterprise, Stash, Team Foundation Server, etc.



Terminology

- Nope nope nope

Exploring and Using GitHub

- Demo!



GitHub

Command Line Git

- Demo!



Further Reading

There's a lot out there. Here are some highlights. Don't be intimidated, you don't need all this, but it may come in handy if you want to dive a bit deeper into the weeds. The key is starting to use it, just like PowerShell!

- Interactive guides
 - GitHub's interactive guide – Learn Git in 15 minutes: <https://try.github.io/>
 - Learn Git Branching: <http://pcottle.github.io/learnGitBranching>
 - Interactive Cheat sheet: <http://ndpssoftware.com/git-cheatsheet.html>
- References
 - Official git reference: <http://git-scm.com/docs>
 - Pro Git (free!): <http://www.git-scm.com/book/en/v2> - the first two or three chapters are a great intro
 - Understanding Branches: <http://blog.thoughtram.io/git/rebase-book/2015/02/10/understanding-branches-in-git.html>
 - Git Explained: For Beginners: <http://www.dotnetcodegeeks.com/2015/06/git-explained-for-beginners.html>
 - GitHub Flow – GitHub from a Browser: <https://github.com/blog/1557-github-flow-in-the-browser>
- Contributing to Microsoft's DSC Resources on GitHub
 - Guide to getting started with GitHub: <https://github.com/PowerShell/DscResources/blob/master/GettingStartedWithGitHub.md>
 - DSC Contributions guide: <https://github.com/PowerShell/DscResources/blob/master/CONTRIBUTING.md>
 - DSC Resource Style Guidelines: <https://github.com/PowerShell/DscResources/blob/master/StyleGuidelines.md>
- More references
 - Pro Git (free!): <http://www.git-scm.com/book/en/v2> - the rest of the book :)
 - A Visual Git Reference: <http://marklodato.github.io/visual-git-guide/index-en.html>
 - Git From the Inside Out: <https://codewords.recurse.com/issues/two/git-from-the-inside-out>
 - Git For Computer Scientists: <http://eagain.net/articles/git-for-computer-scientists> - Great read with pictures, don't be intimidated by the title
 - Branching, forking, other concepts explained: <http://stackoverflow.com/questions/3329943/git-branch-fork-fetch-merge-rebase-and-clone-what-are-the-differences>
- Oh shoot, something went wrong!
 - So you have a mess on your hands workflow: <http://justinhileman.info/article/git-pretty/git-pretty.png>
 - How to undo (almost) anything with Git: <https://github.com/blog/2019-how-to-undo-almost-anything-with-git>
- A few recent articles:
 - Git Disciplined: <http://blog.8thlight.com/makis-otman/2015/07/08/git-disciplined.html> - Thanks for sharing Steve!
 - Git for IT Professionals: Getting Started: <http://www.powershellmagazine.com/2015/07/13/git-for-it-professionals-getting-started-2/> - Thanks to Ravikanth! More to come in this series

בואו ננסה

- Git animation: <http://pel-daniel.github.io/git-init/>
- Learn git app: <https://github.com/jlord/git-it-electron/releases>
- Docs: <https://docs.github.com/en/get-started/quickstart>
 - (Bonus: try: <https://education.github.com/pack>)
- Lab:
<https://lab.github.com/githubtraining/introduction-to-github>



משל גיט

- Tom Preston-Werner
<http://tom.preston-werner.com/2009/05/19/the-git-parable.html>
- Herland, <http://www.infoq.com/presentations/git-details>,
slides: https://github.com/jherland/git_parable

The Git Parable

Johan Herland

johan@herland.net

The Git Parable

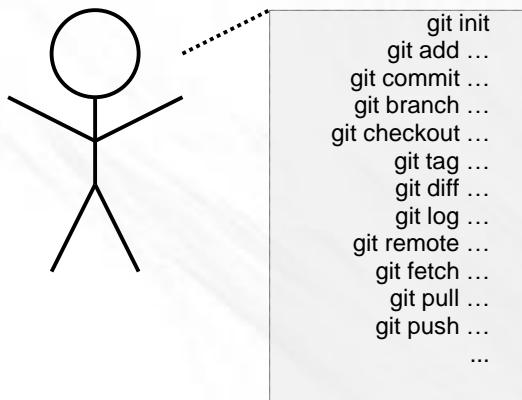
- Shamelessly stolen from Tom Preston-Werner
<http://tom.preston-werner.com/2009/05/19/the-git-parable.html>
- I'm lazy...
- Also: Best introduction to Git I've found so far

The Git Parable

- Git - simple & powerful

The Git Parable

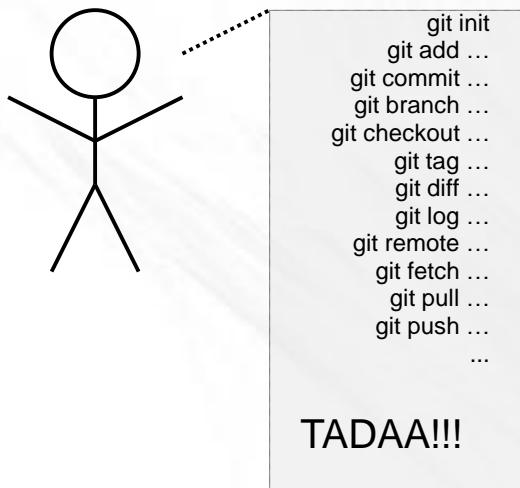
- Git - simple & powerful



git init
git add ...
git commit ...
git branch ...
git checkout ...
git tag ...
git diff ...
git log ...
git remote ...
git fetch ...
git pull ...
git push ...
...

The Git Parable

- Git - simple & powerful



The Git Parable

- Git - simple & powerful



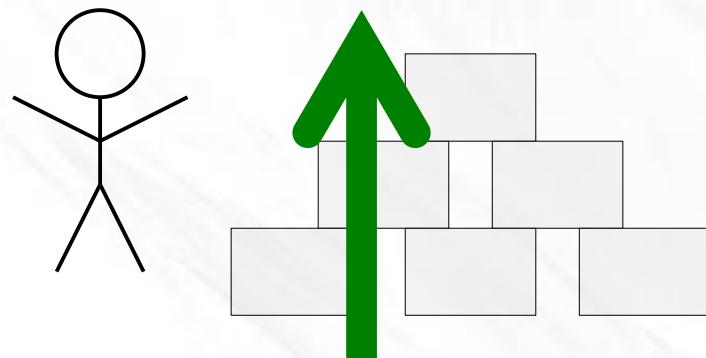
The Git Parable

- Git - simple & powerful



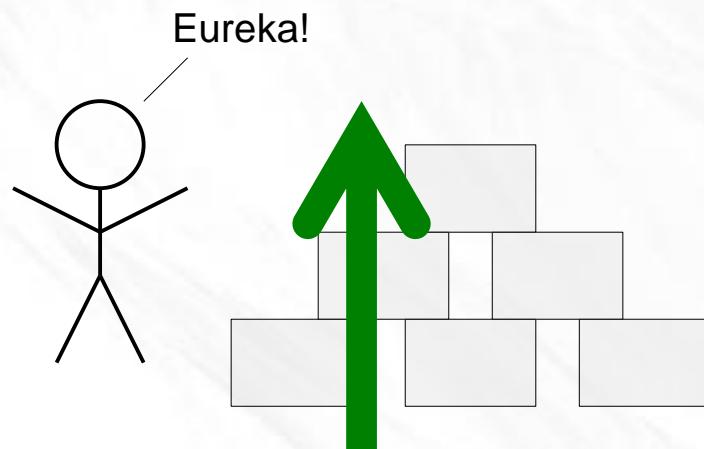
The Git Parable

- Git - simple & powerful



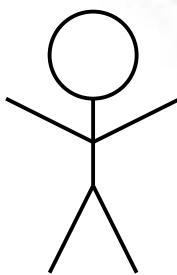
The Git Parable

- Git - simple & powerful



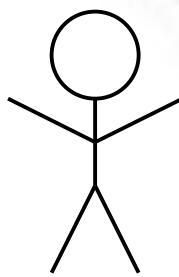
The Parable

- A simple computer
 - A text editor
 - A few filesystem commands



The Parable

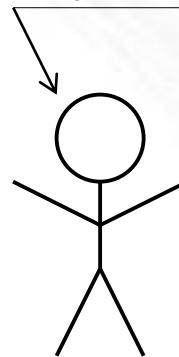
- Write a large software program



The Parable

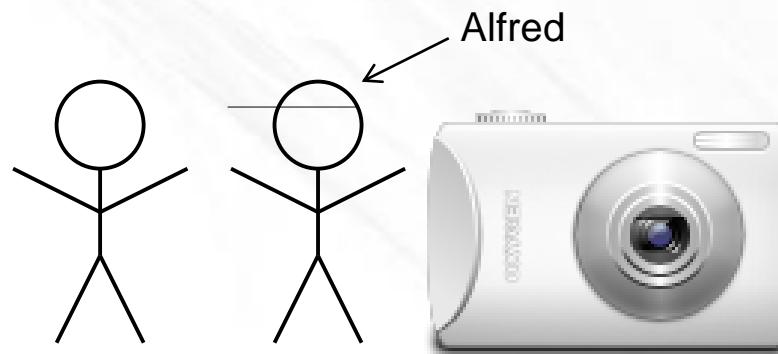
- Write a large software program
- Invent some method to keep track of versions
 - retrieve code that you changed/deleted

Responsible!



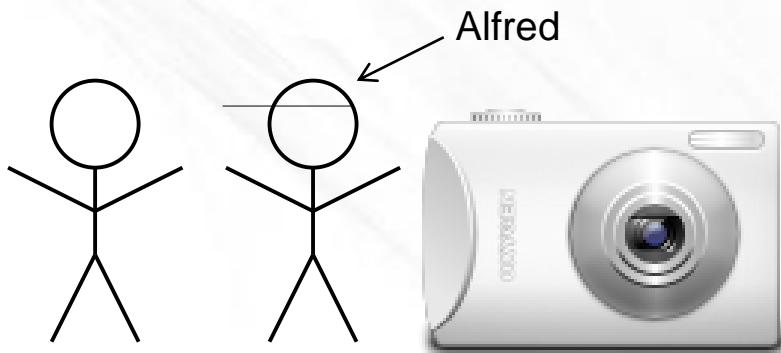
Snapshots

- Alfred, the photographer



Snapshots

- Alfred, the photographer



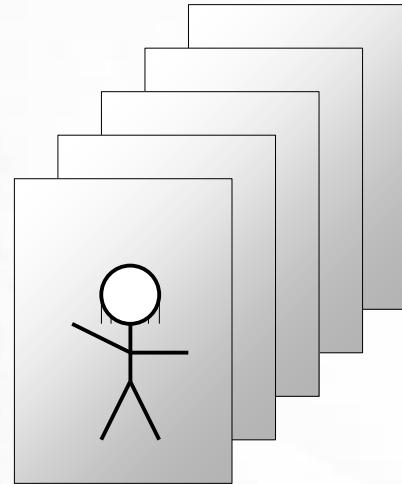
Snapshots

- Alfred, the photographer
- Hazel and her daughter



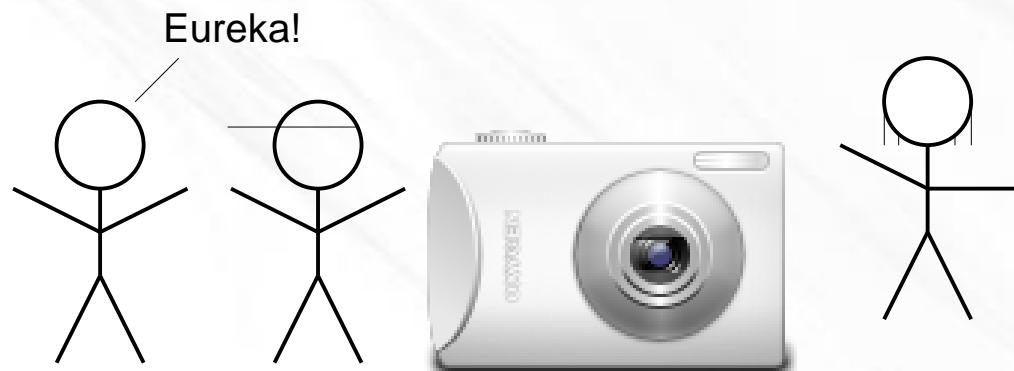
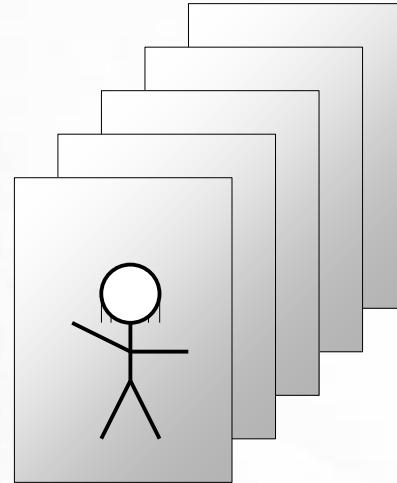
Snapshots

- Alfred, the photographer
- Hazel and her daughter
 - Remember what the daughter was like at each different stage



Snapshots

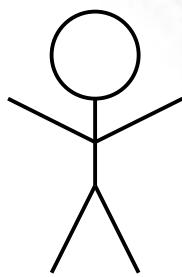
- Alfred, the photographer
- Hazel and her daughter
 - Remember what the daughter was like at each different stage



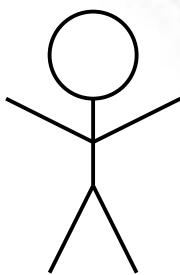
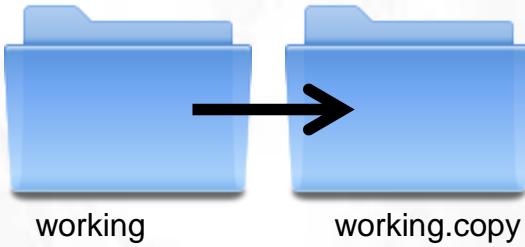
Snapshots



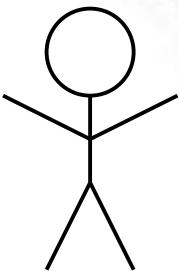
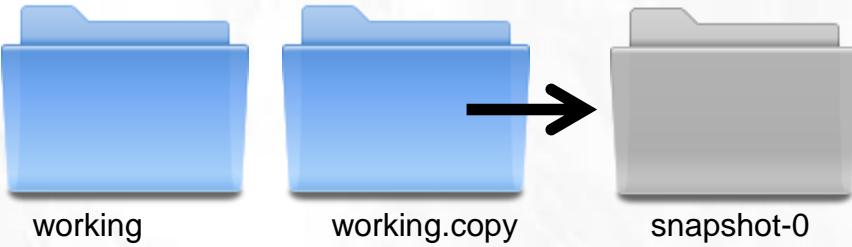
working



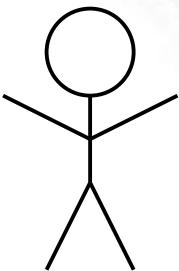
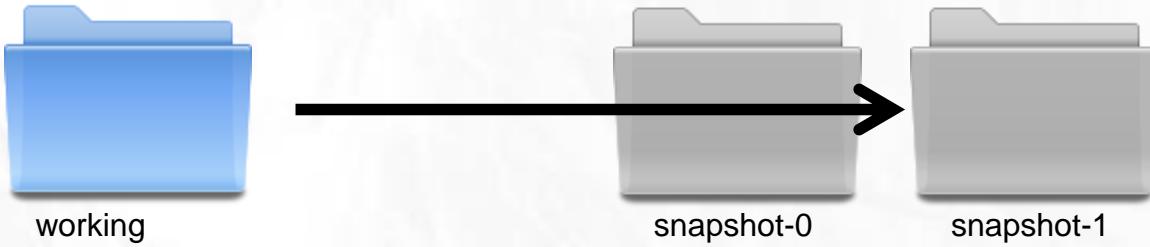
Snapshots



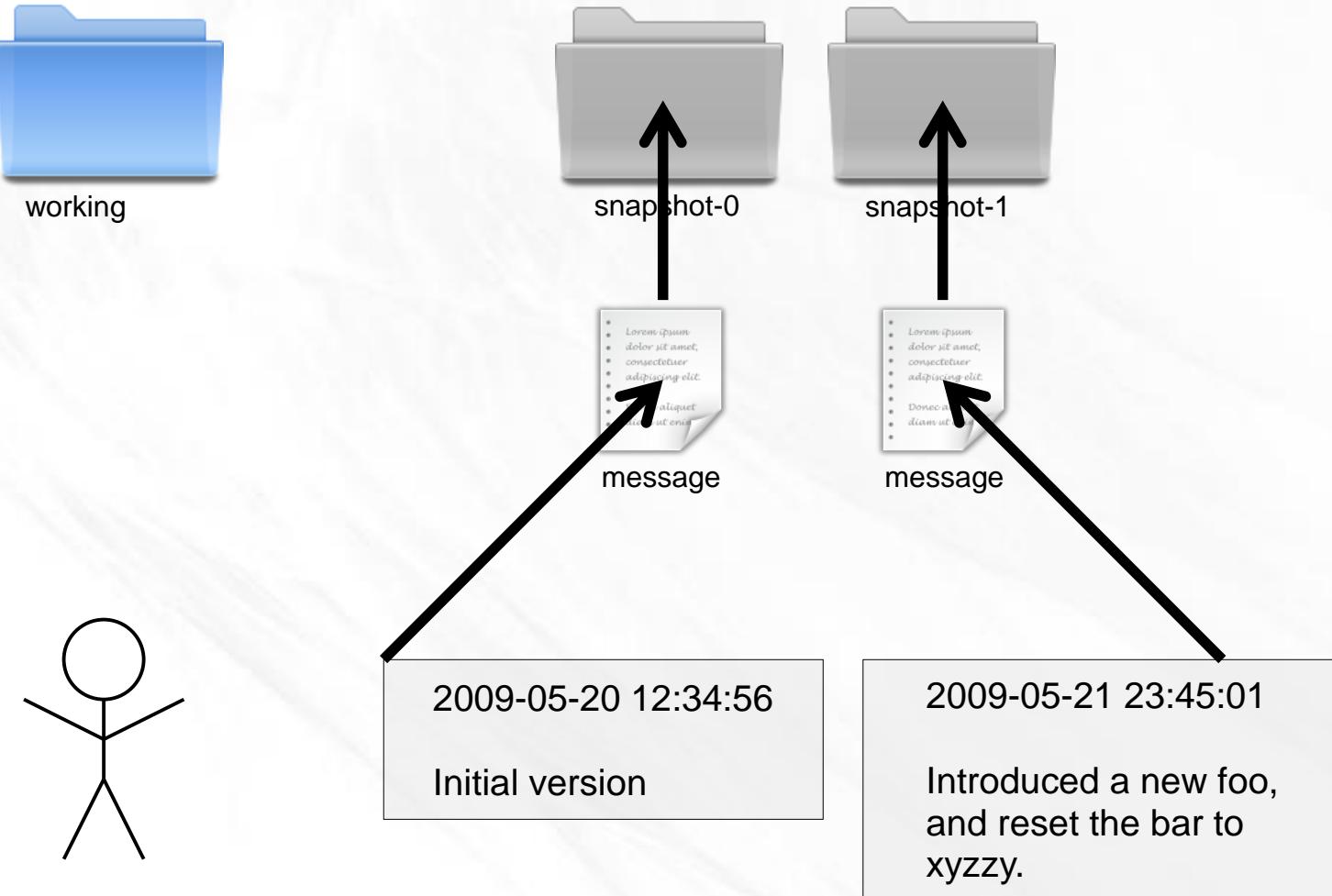
Snapshots



Snapshots



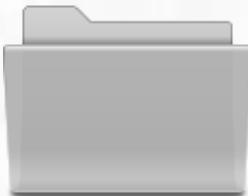
Snapshots



Branches



working



snapshot-0

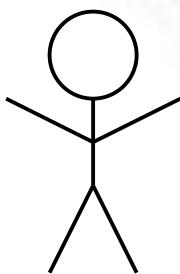


snapshot-1

...



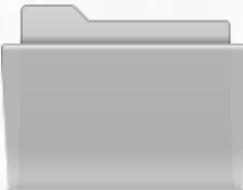
snapshot-99



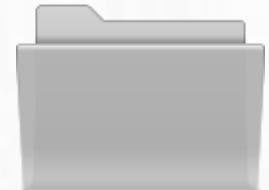
Branches



working



snapshot-0

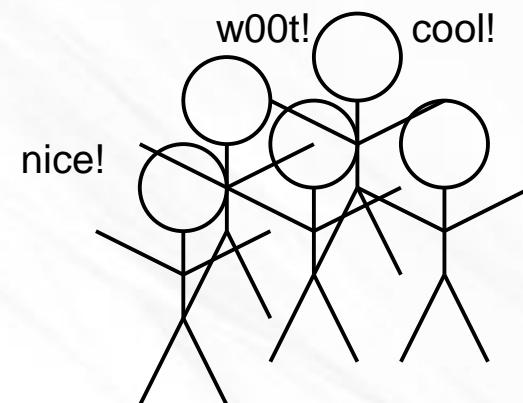
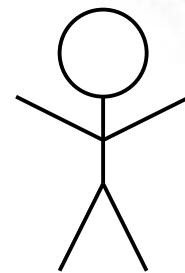


snapshot-1

...



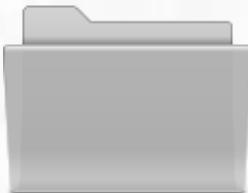
snapshot-99



Branches



working

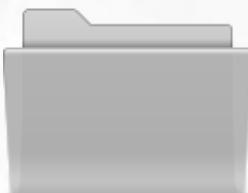


snapshot-0



snapshot-1

...

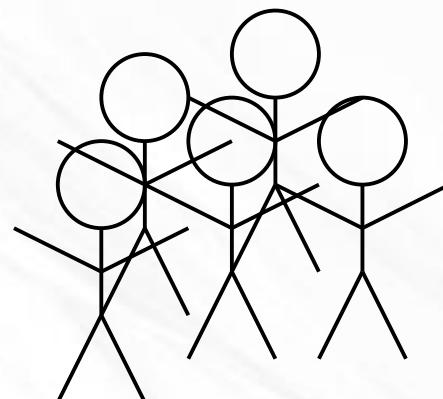
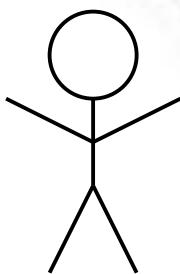


snapshot-99



snapshot-109

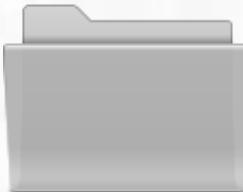
...



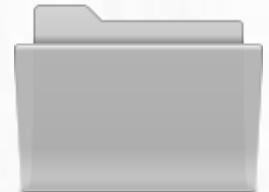
Branches



working



snapshot-0



snapshot-1

...



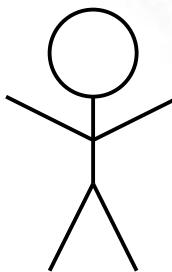
snapshot-99



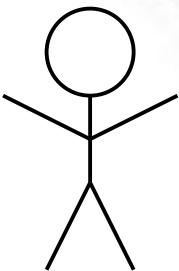
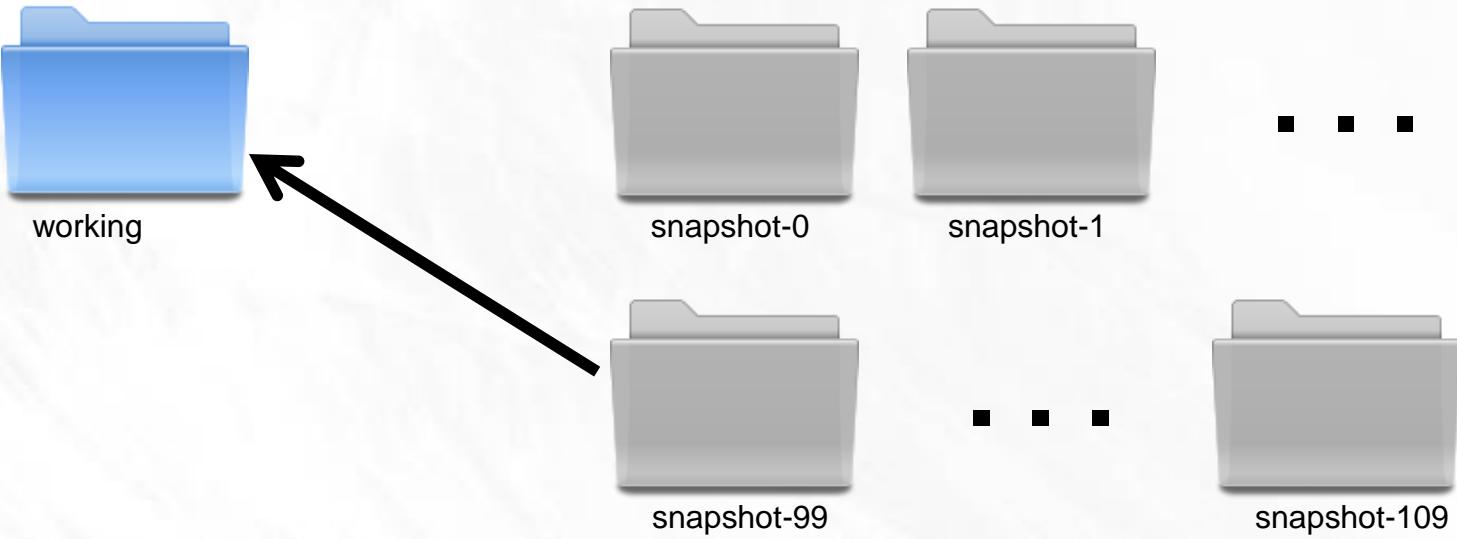
snapshot-109

...

Does not work!



Branches



Branches



working



snapshot-0



snapshot-1

...



snapshot-99

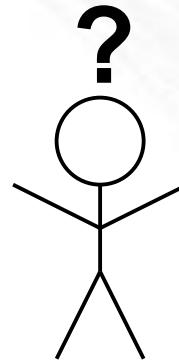


snapshot-109

...



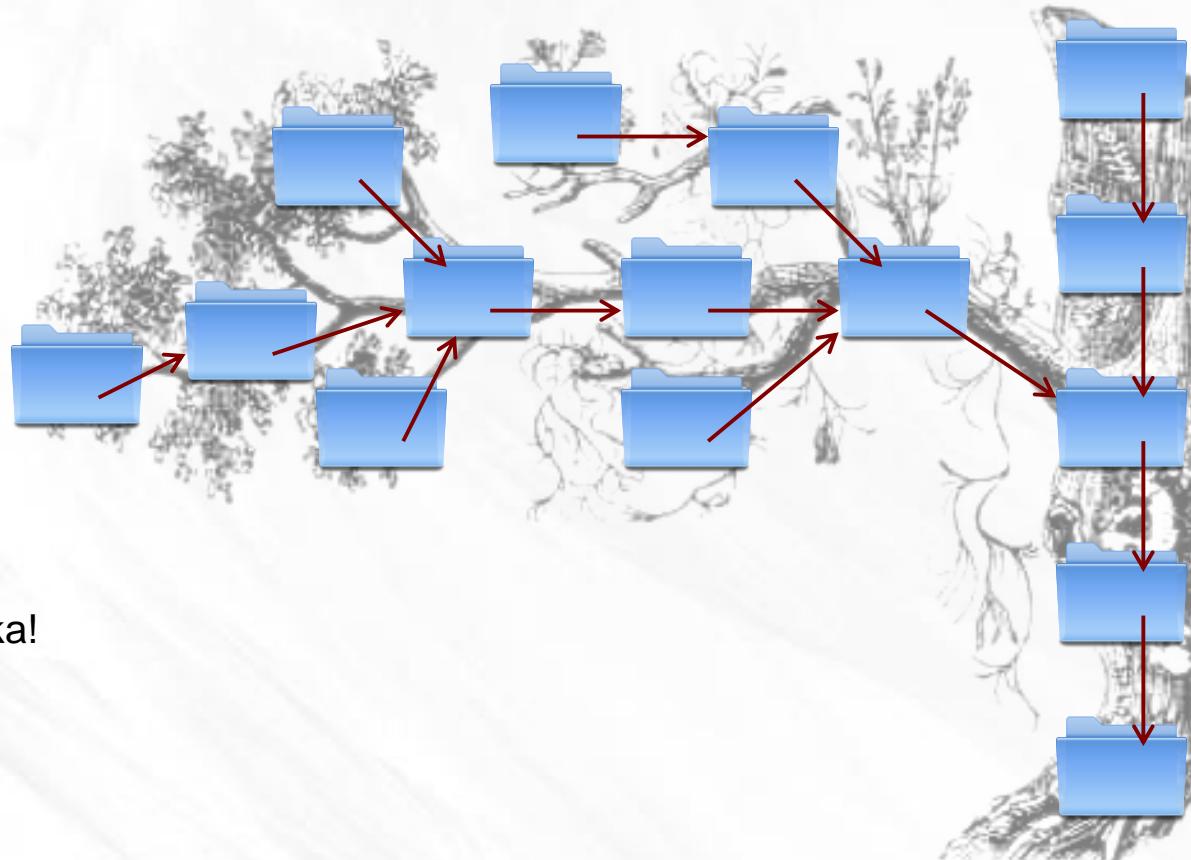
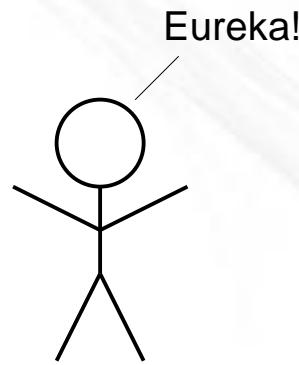
snapshot-110



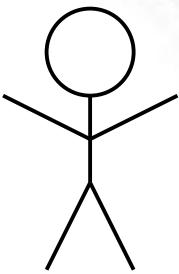
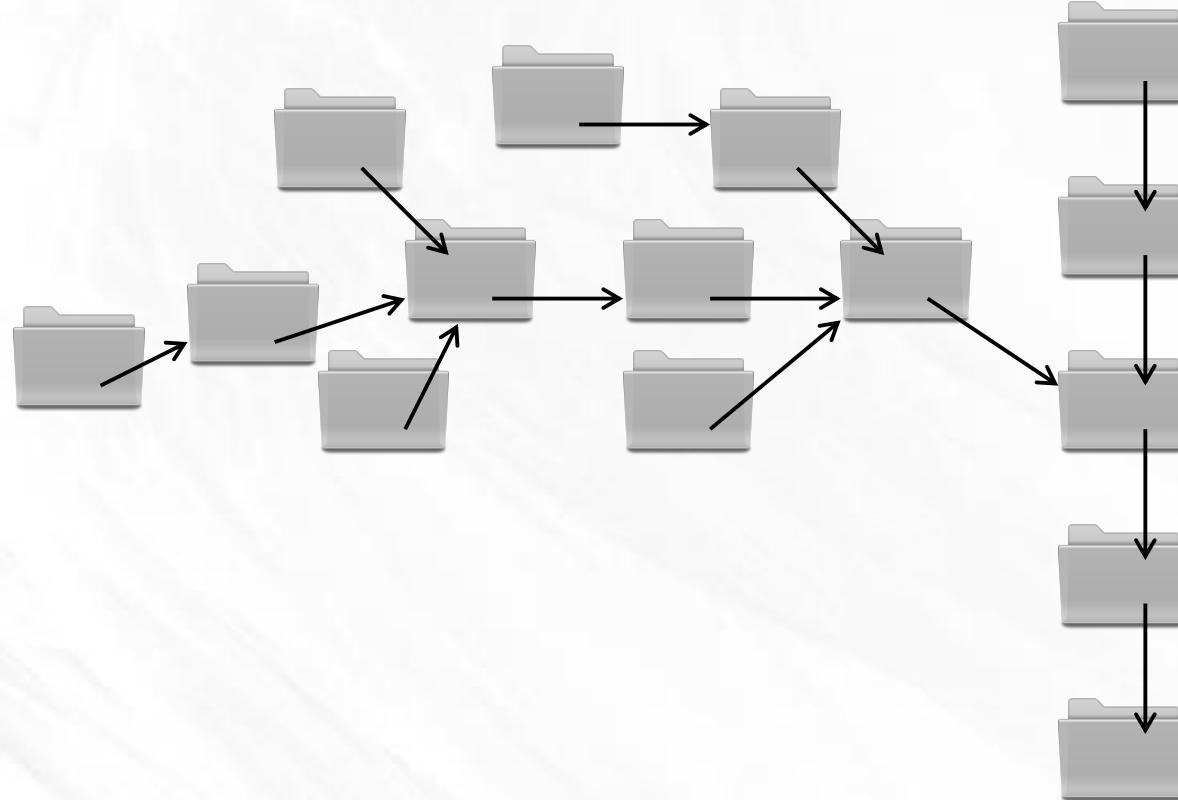
Branches



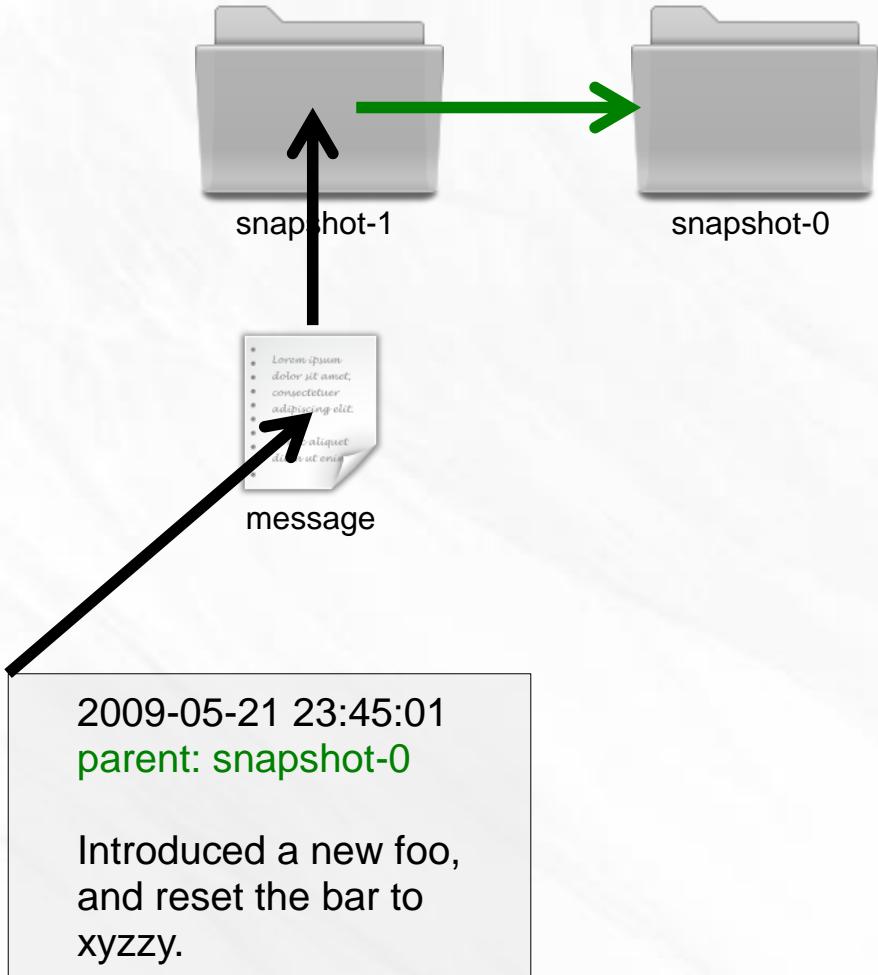
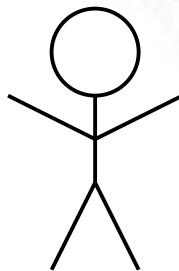
Branches



Branches



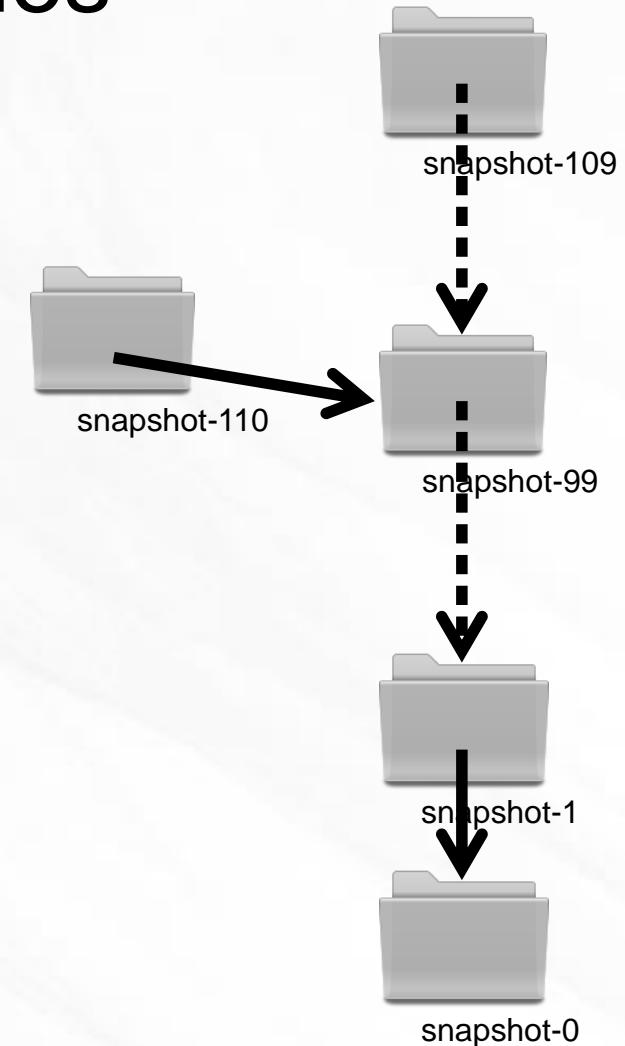
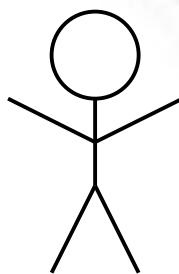
Branches



Branch Names



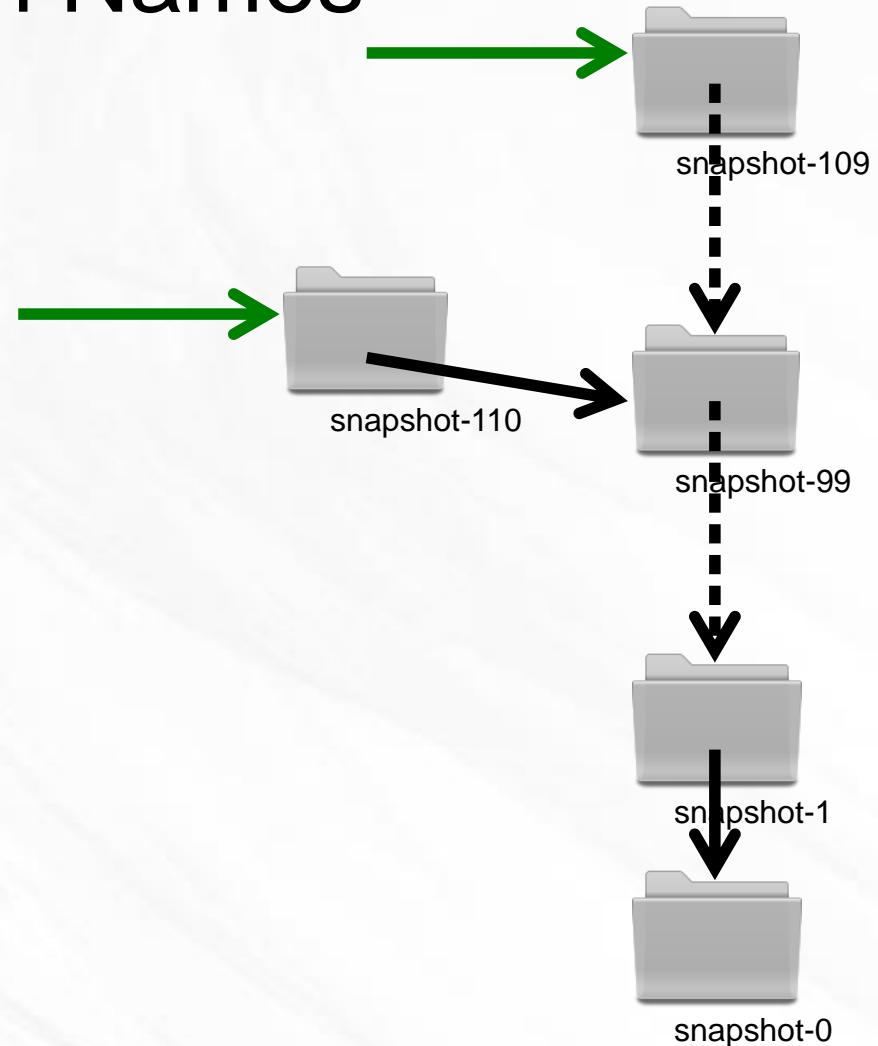
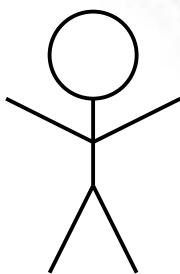
working



Branch Names



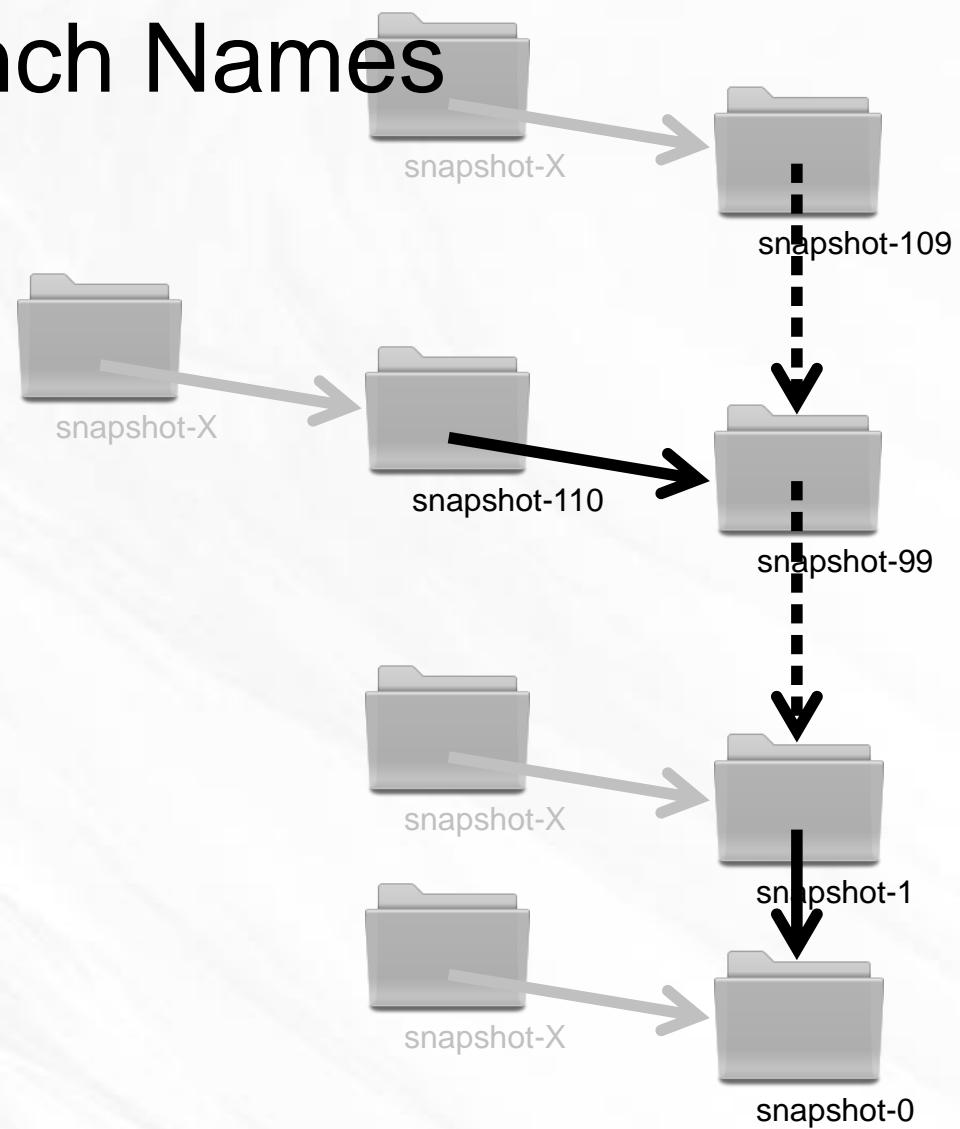
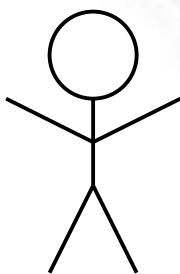
working



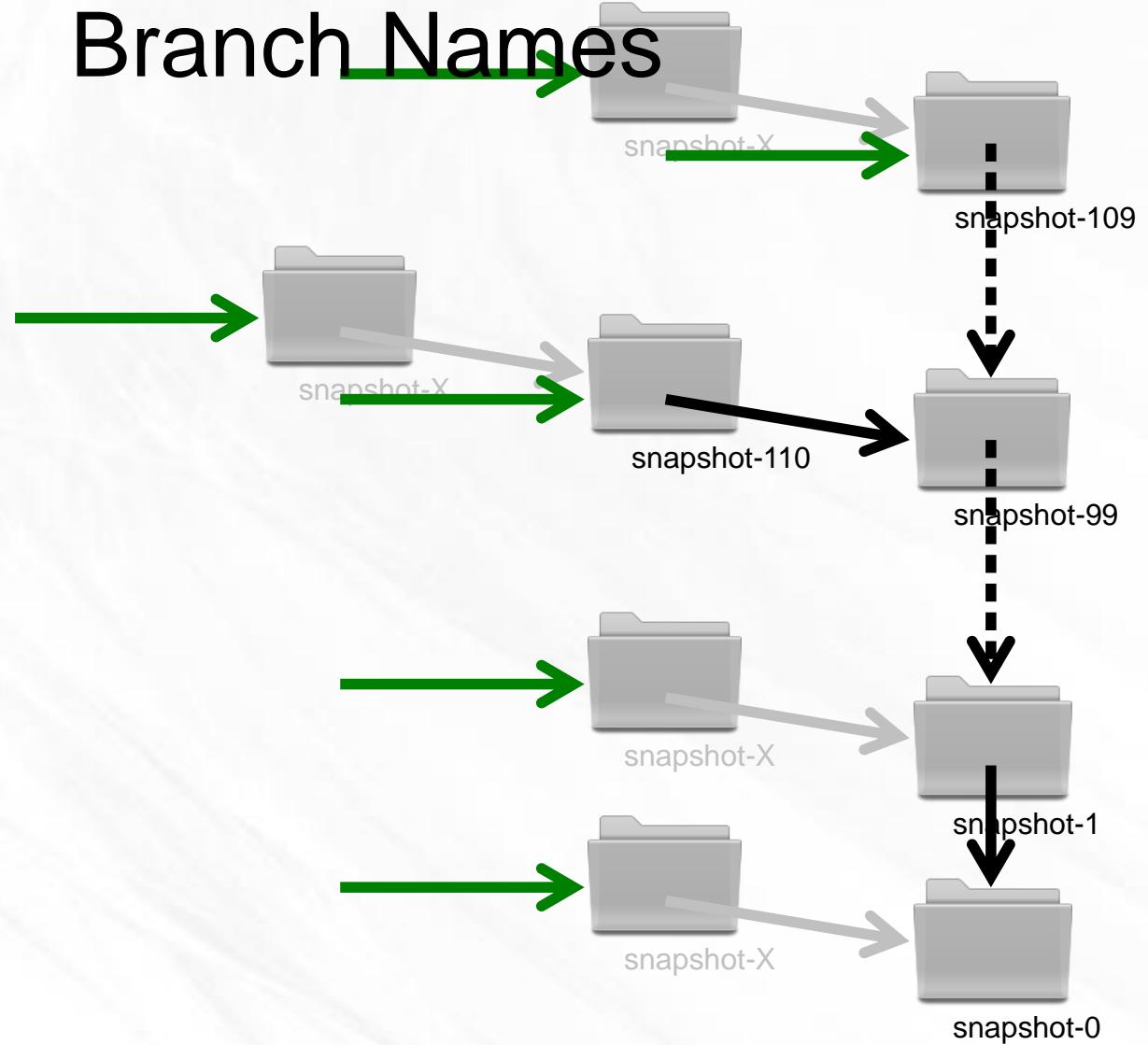
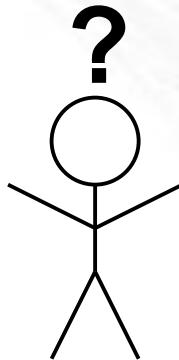
Branch Names



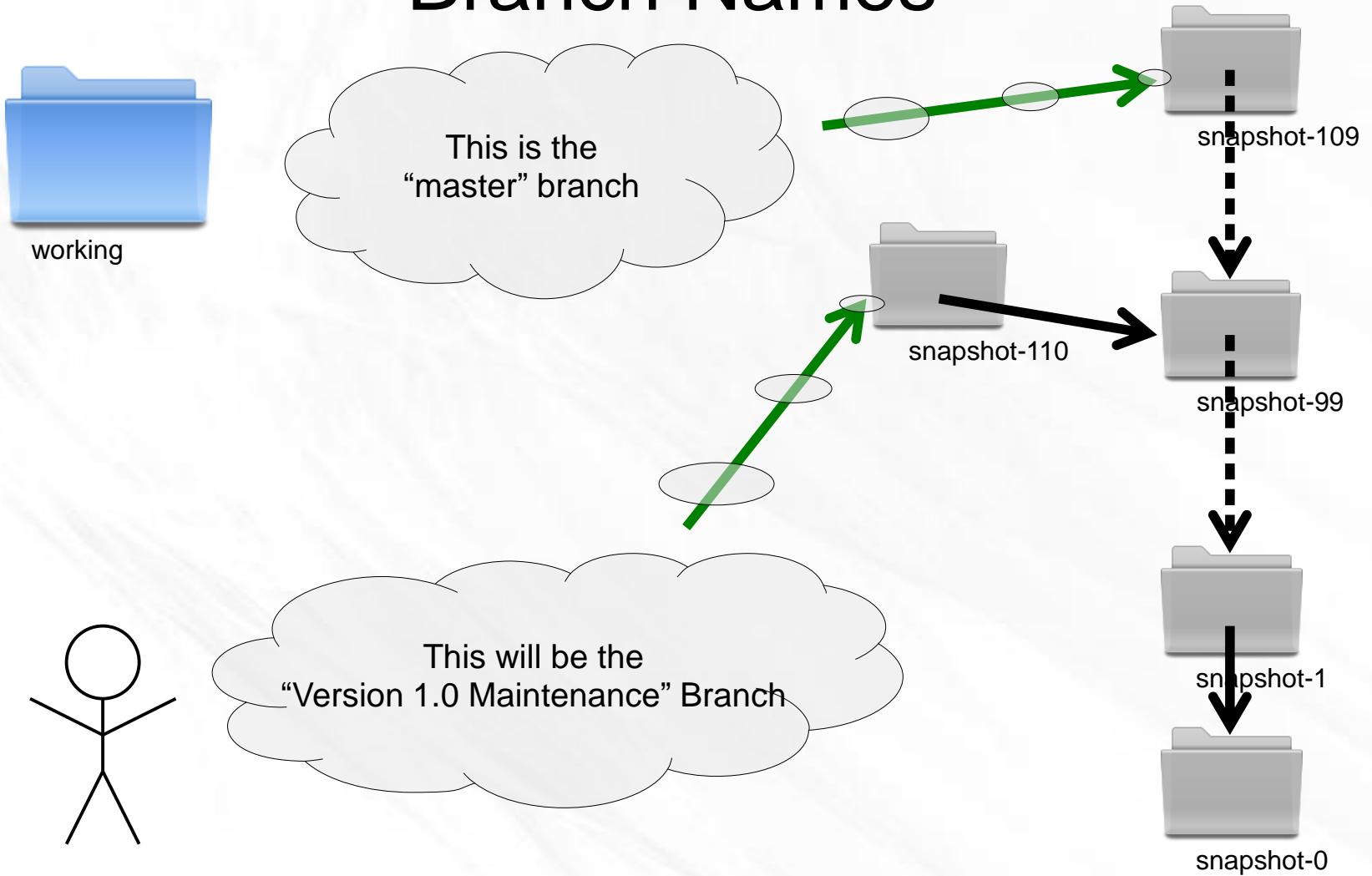
working



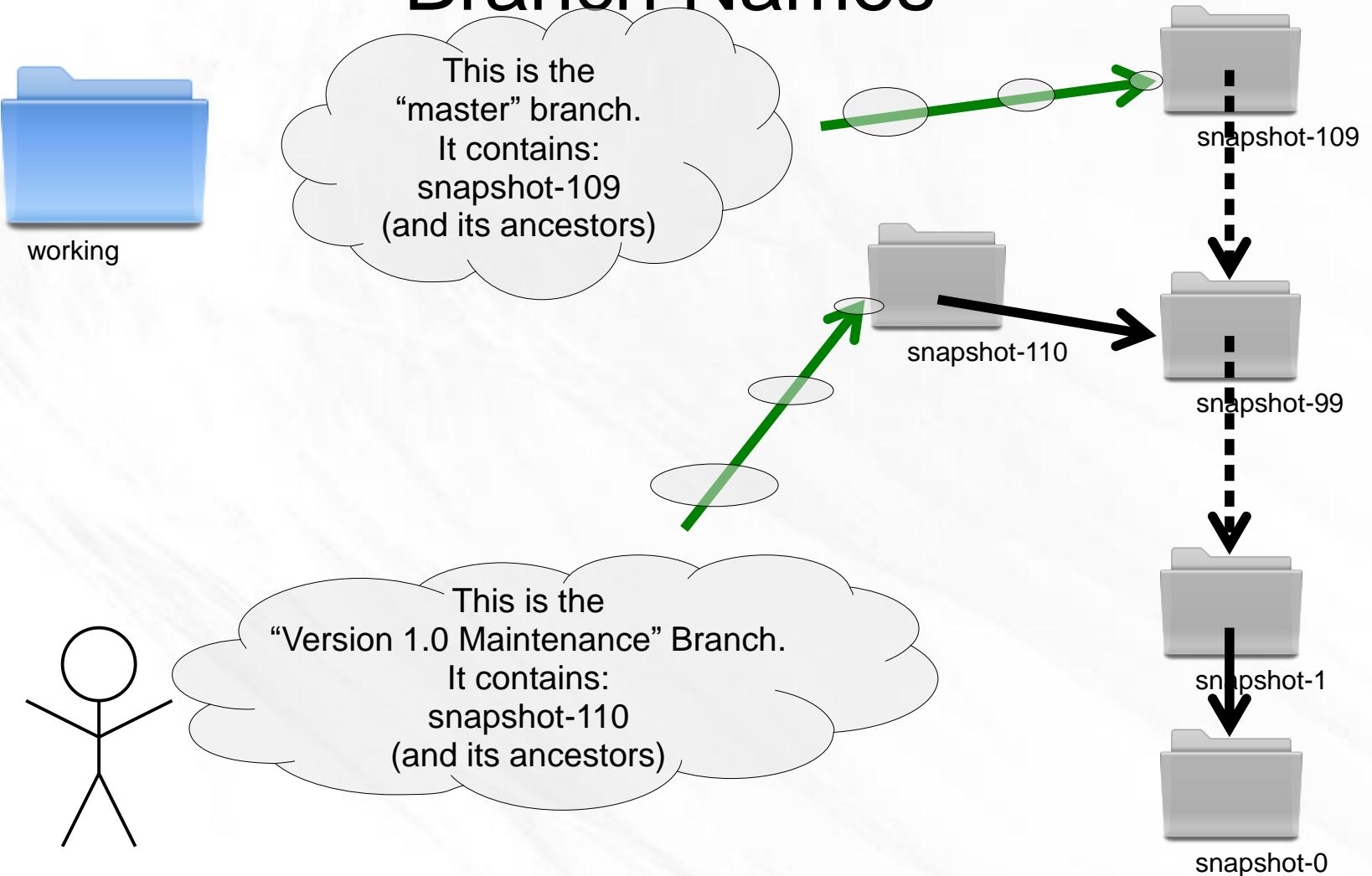
Branch Names



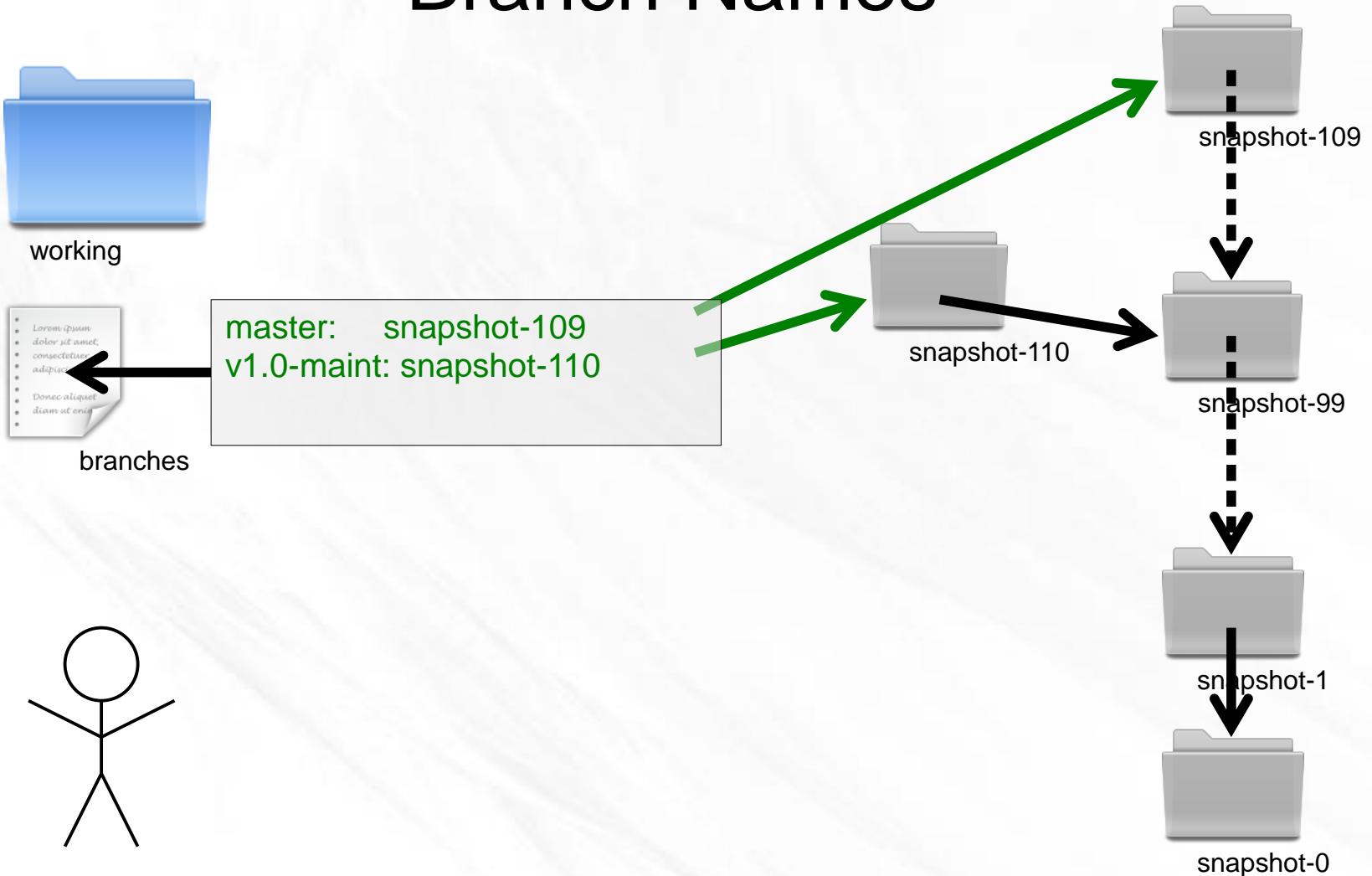
Branch Names



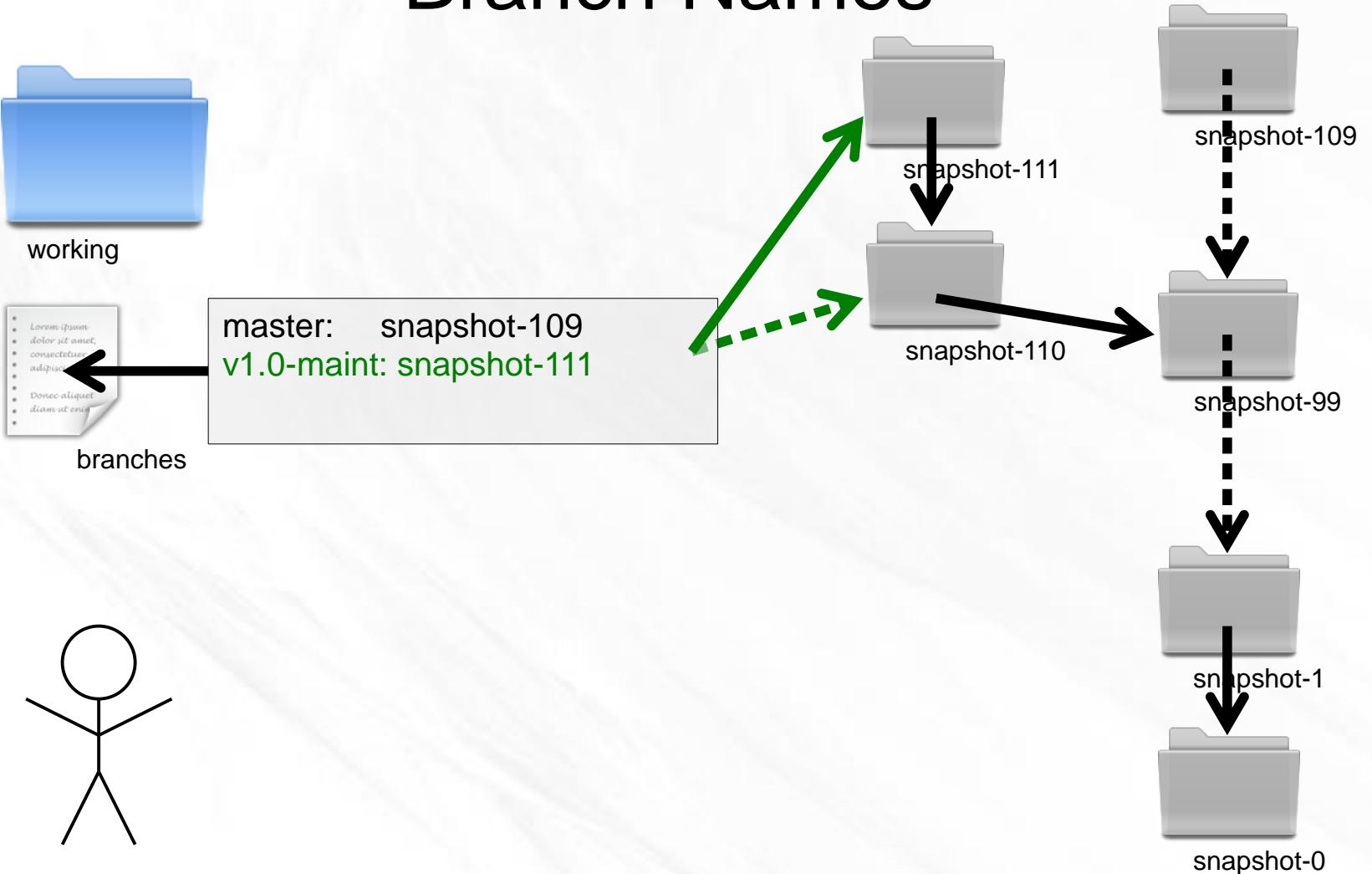
Branch Names



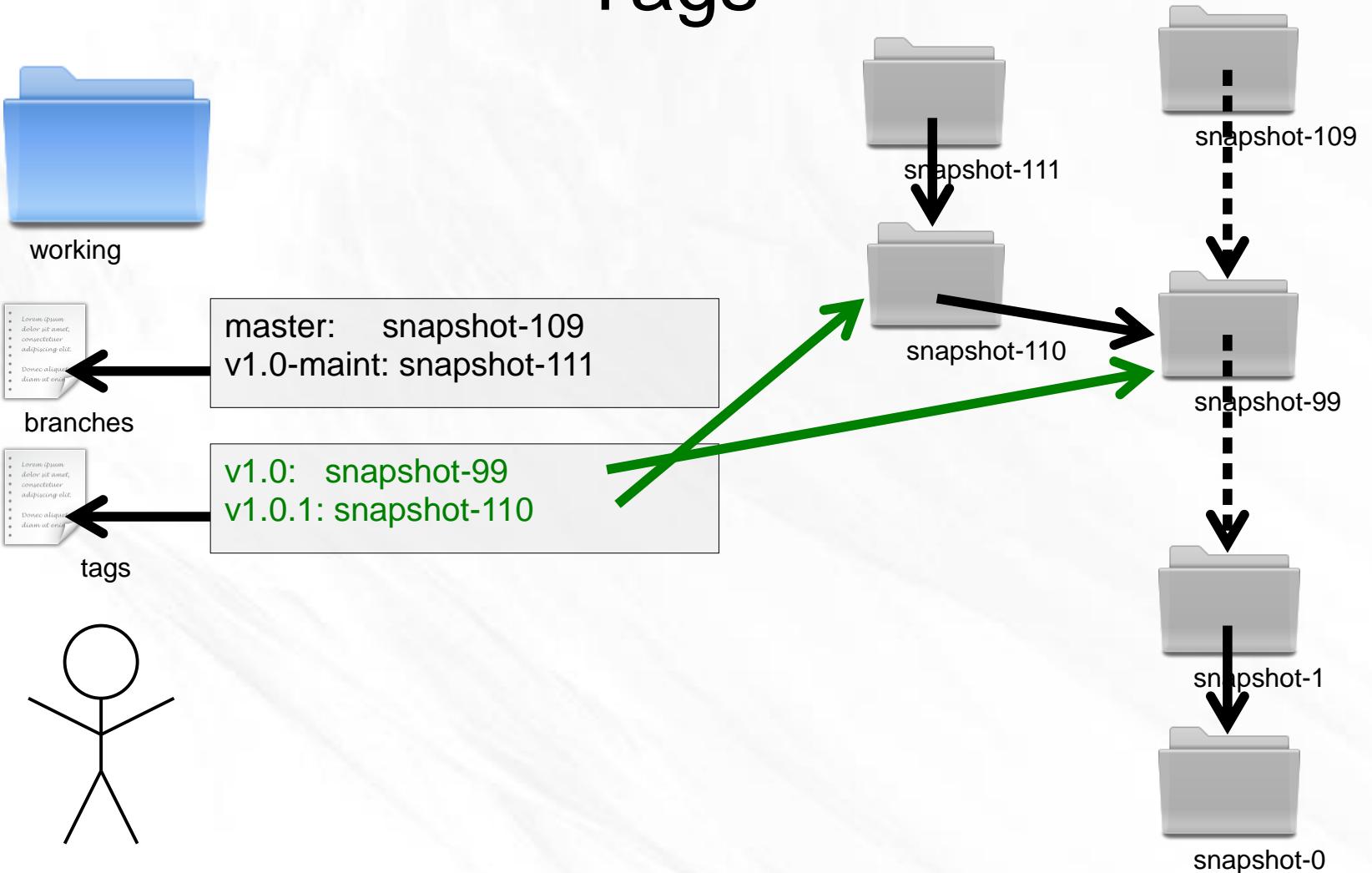
Branch Names



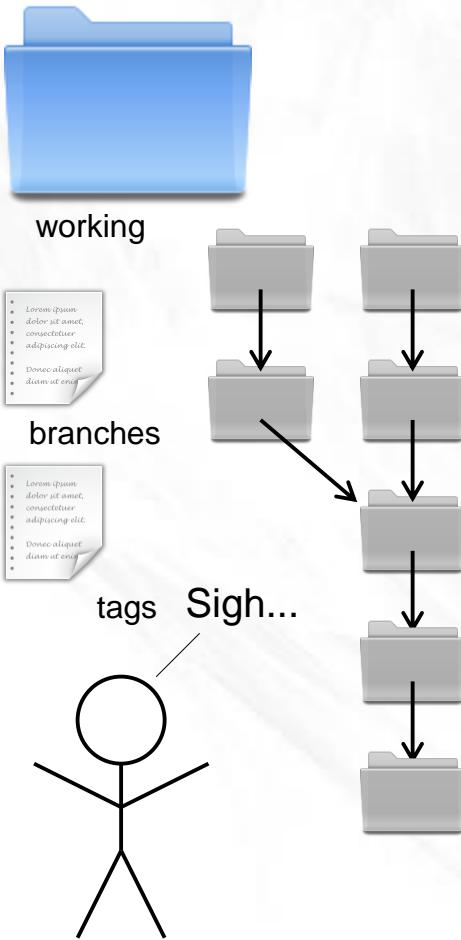
Branch Names



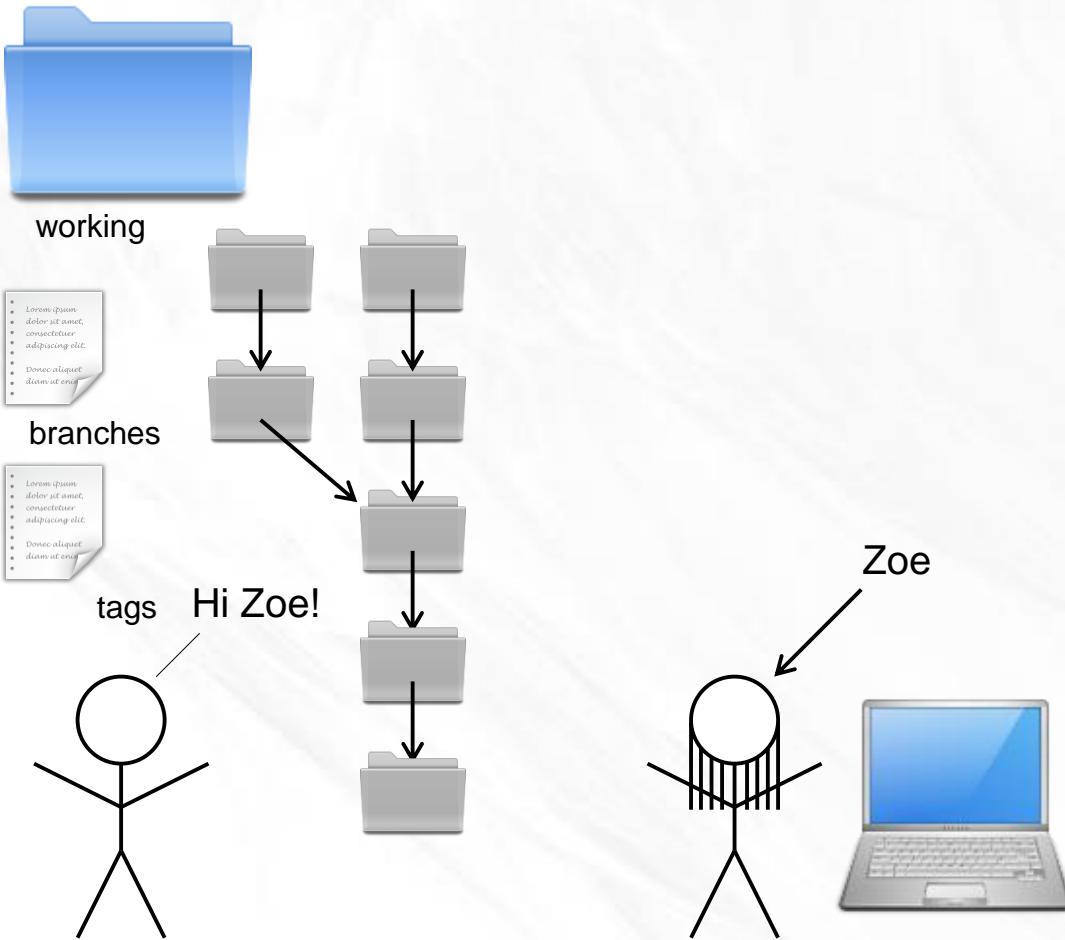
Tags



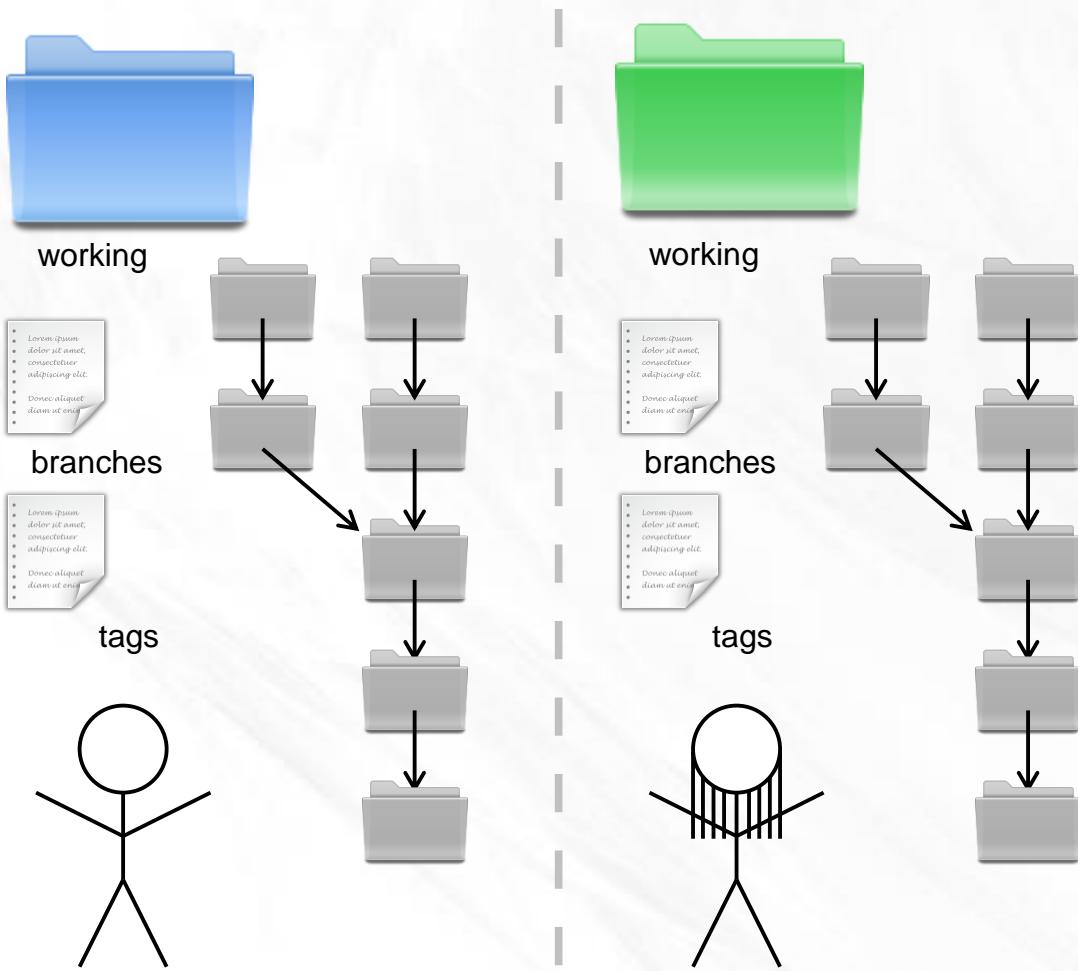
Distributed



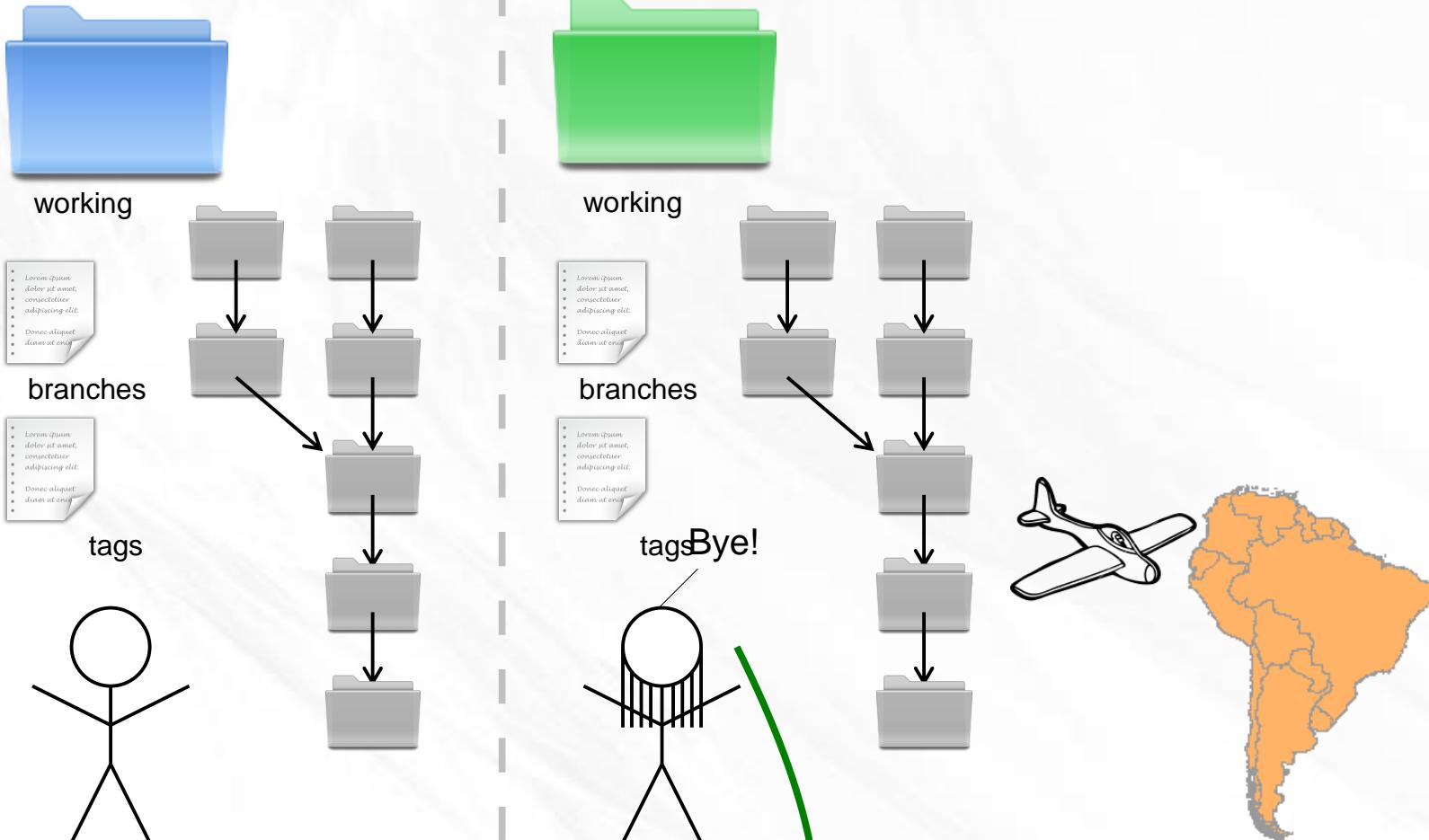
Distributed



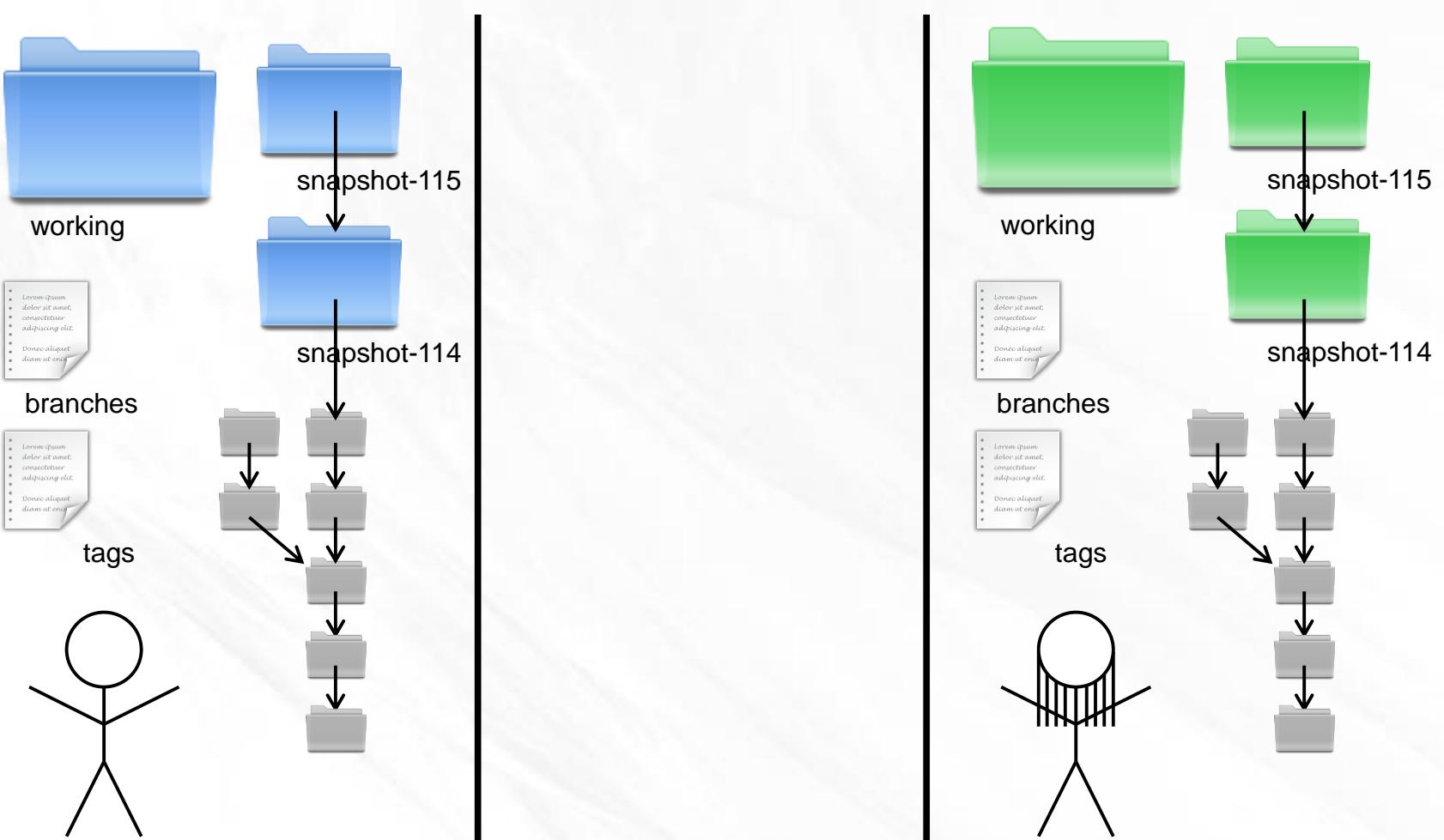
Distributed



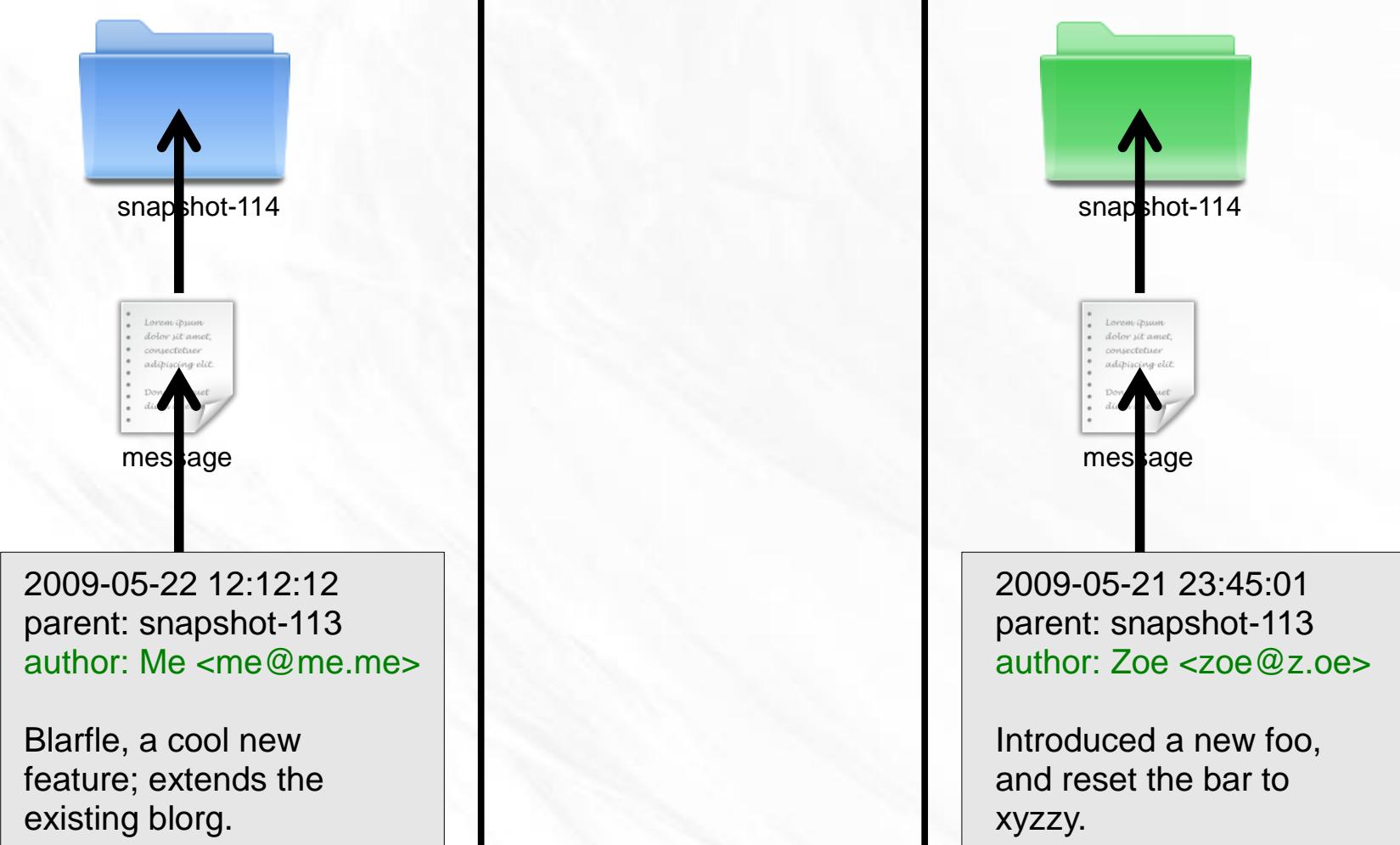
Distributed



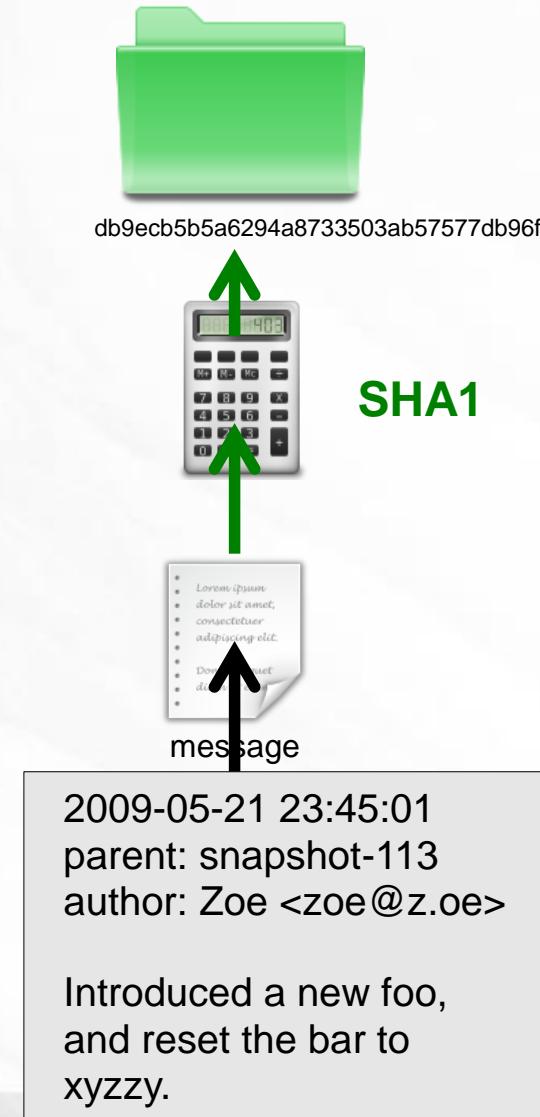
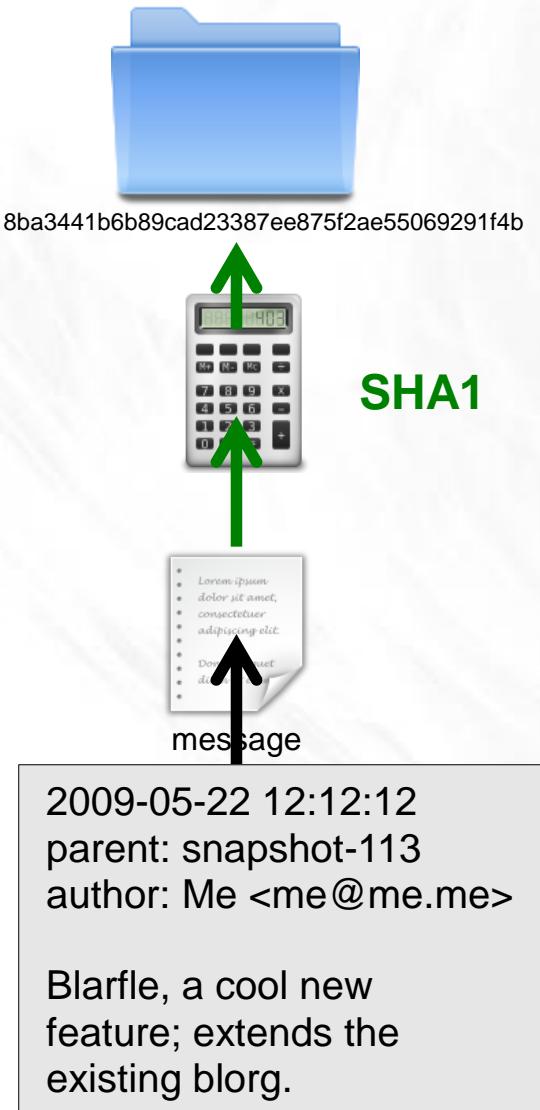
Distributed



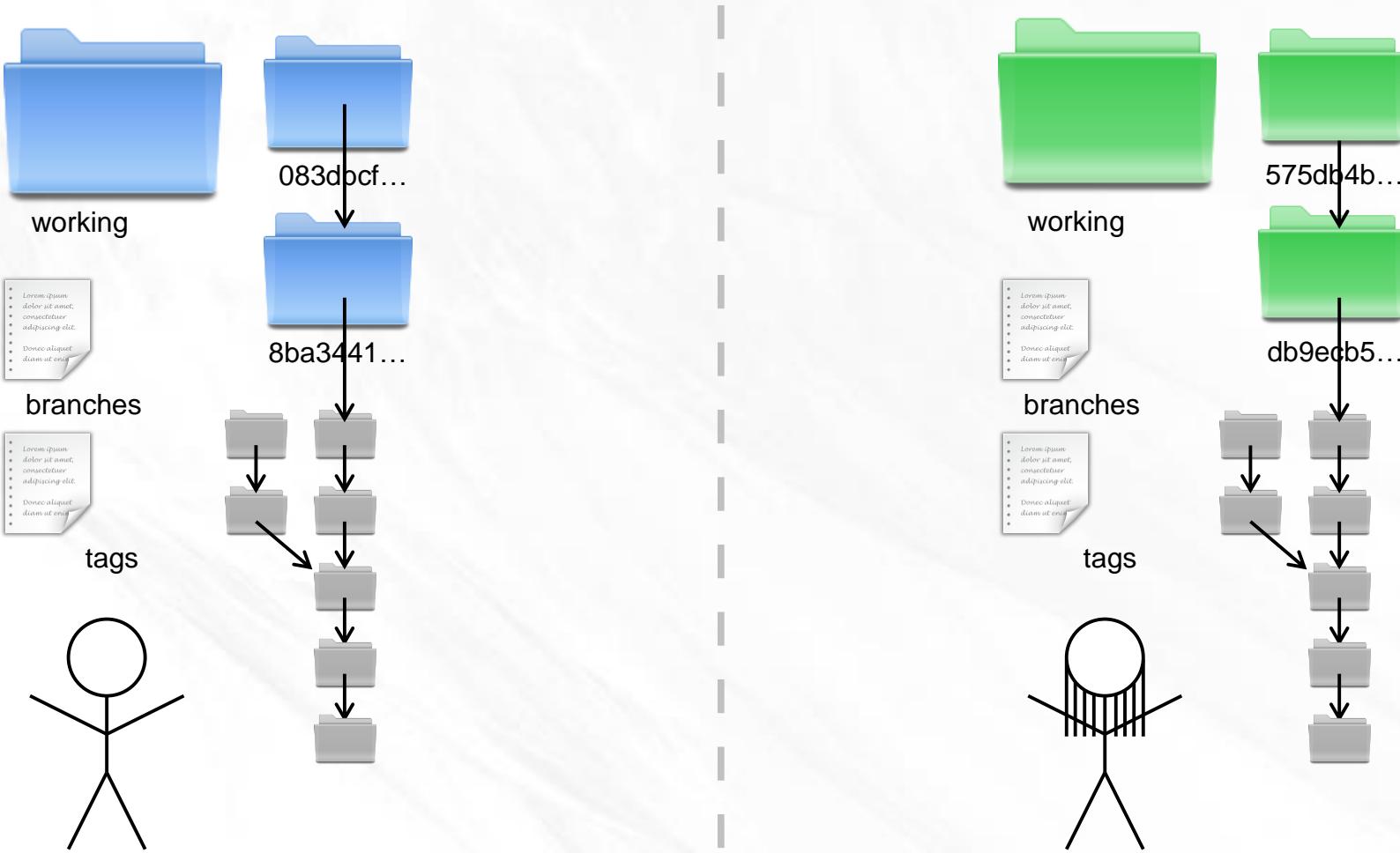
Distributed



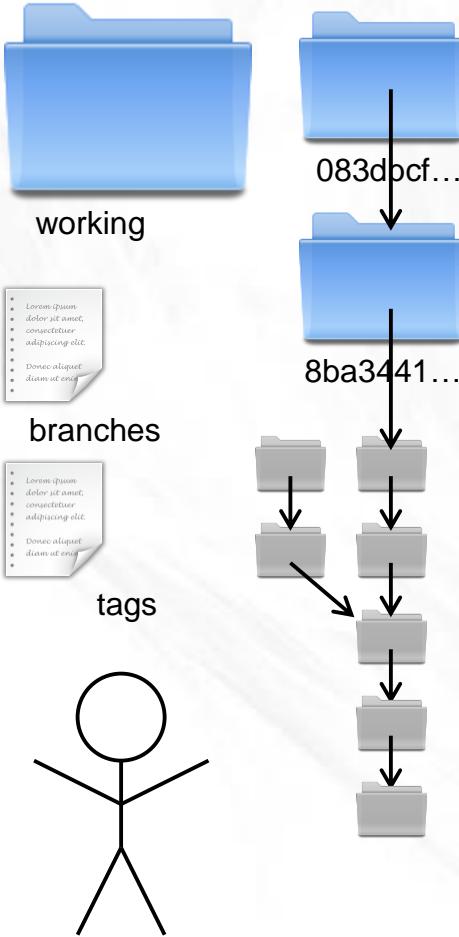
Distributed



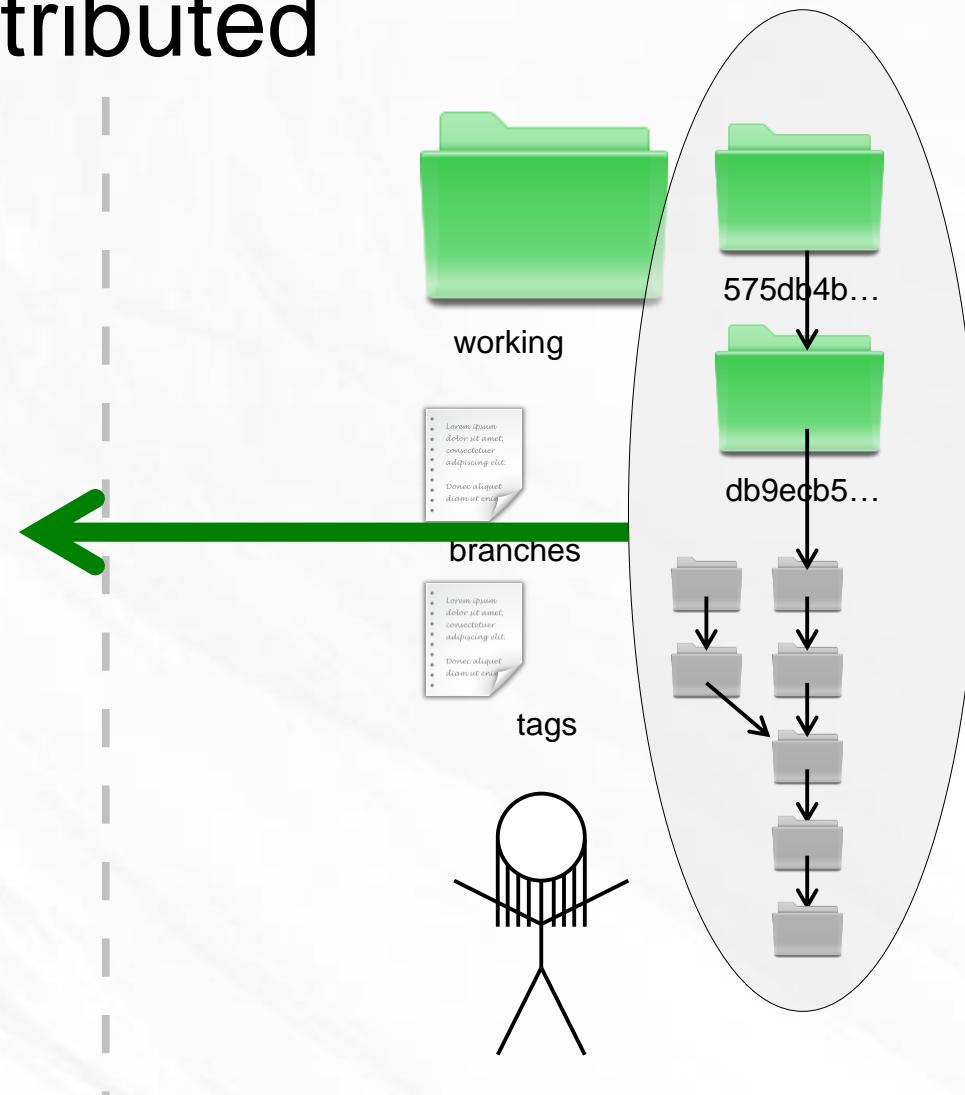
Distributed



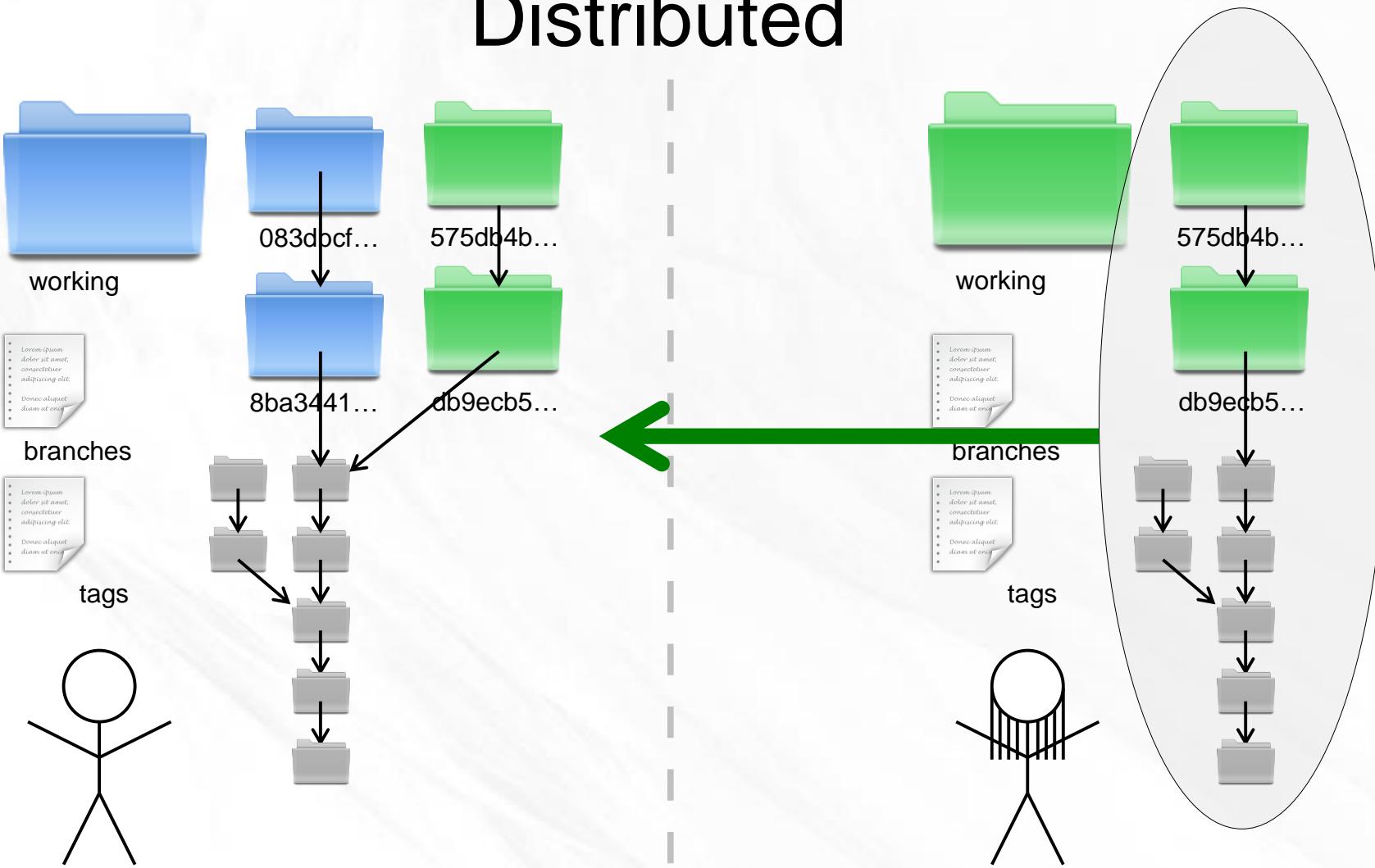
Distributed



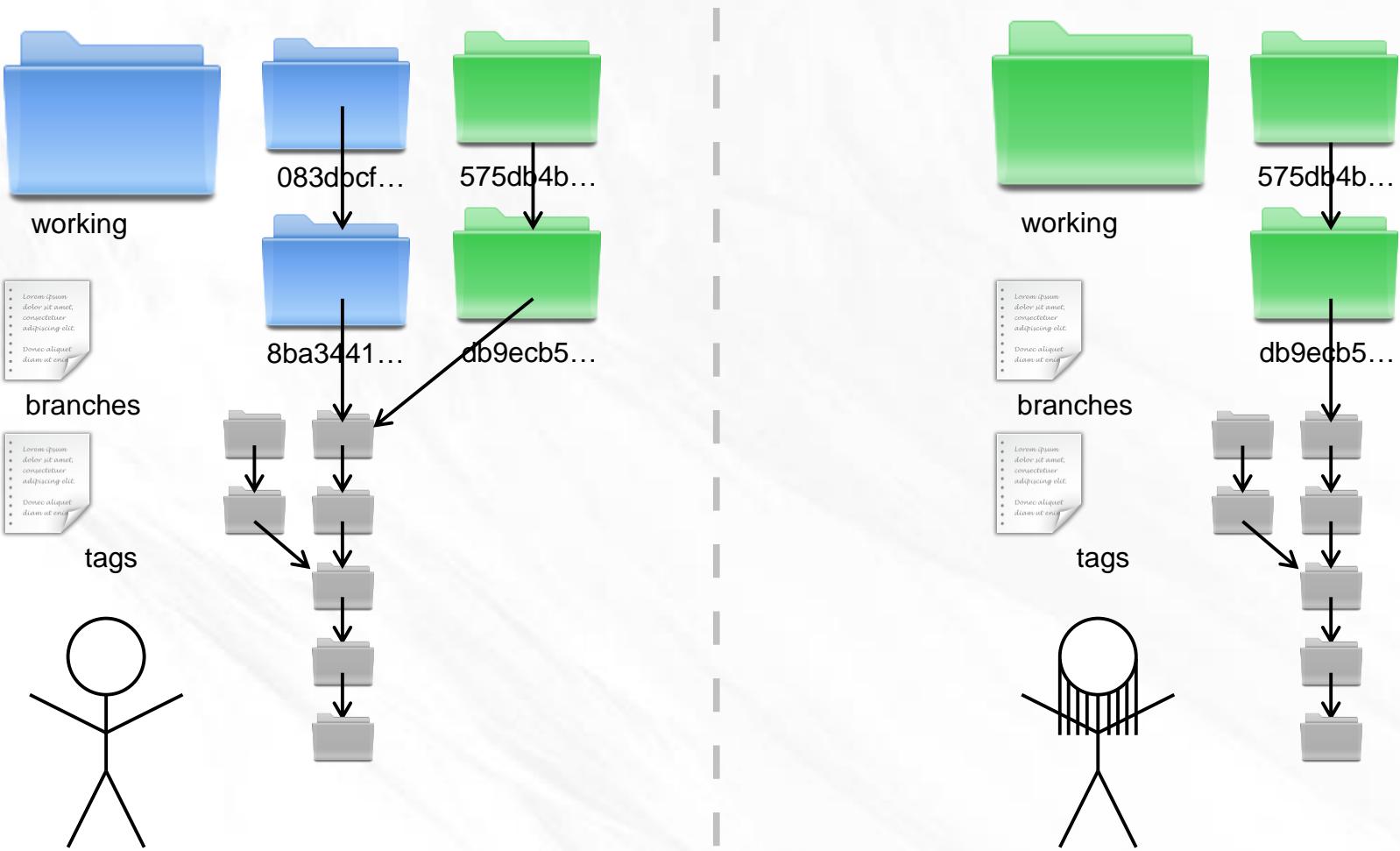
?



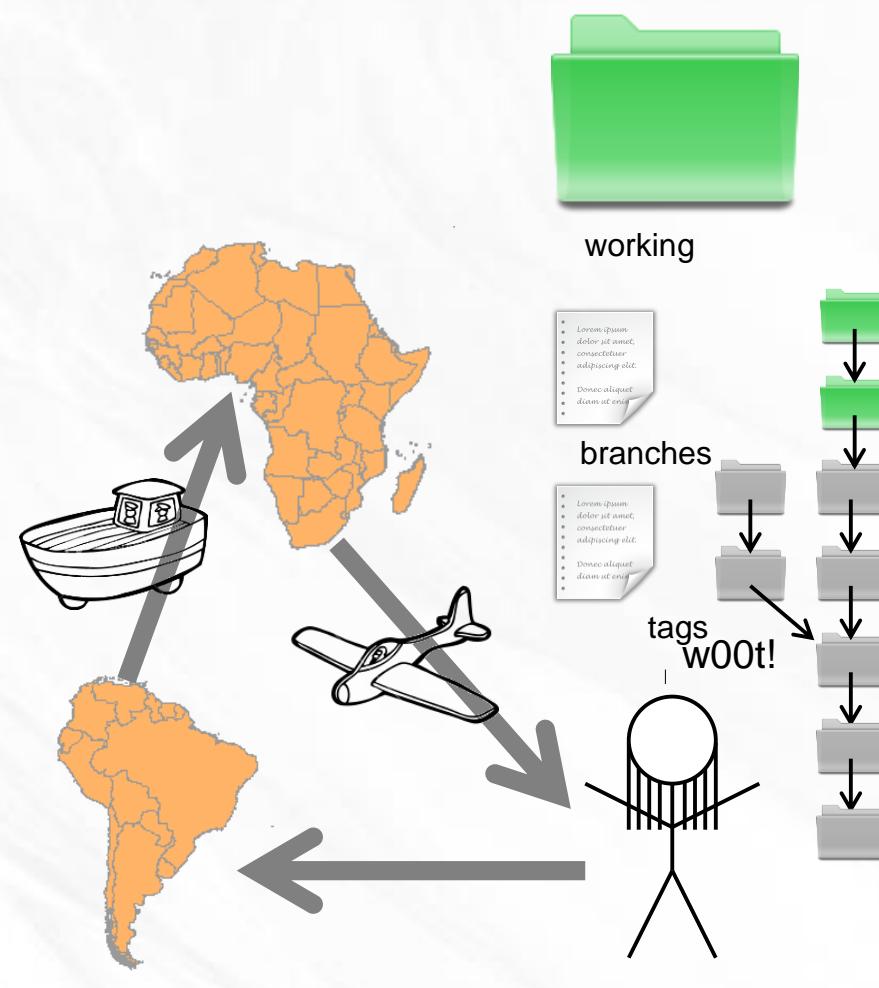
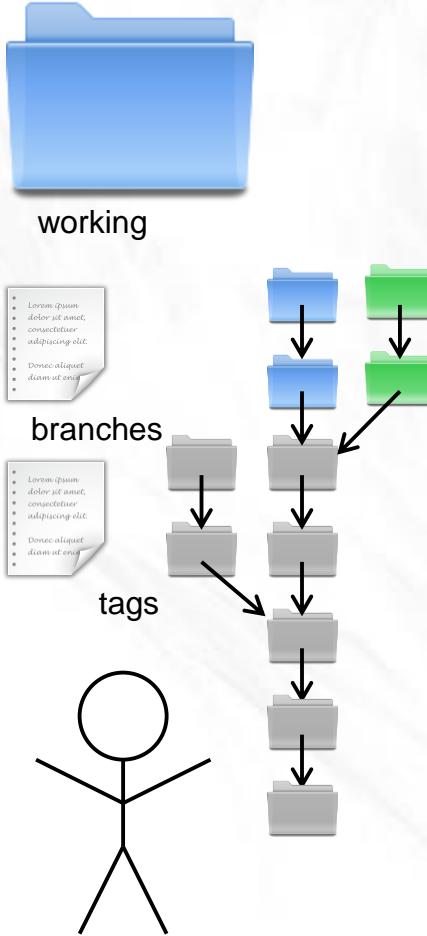
Distributed



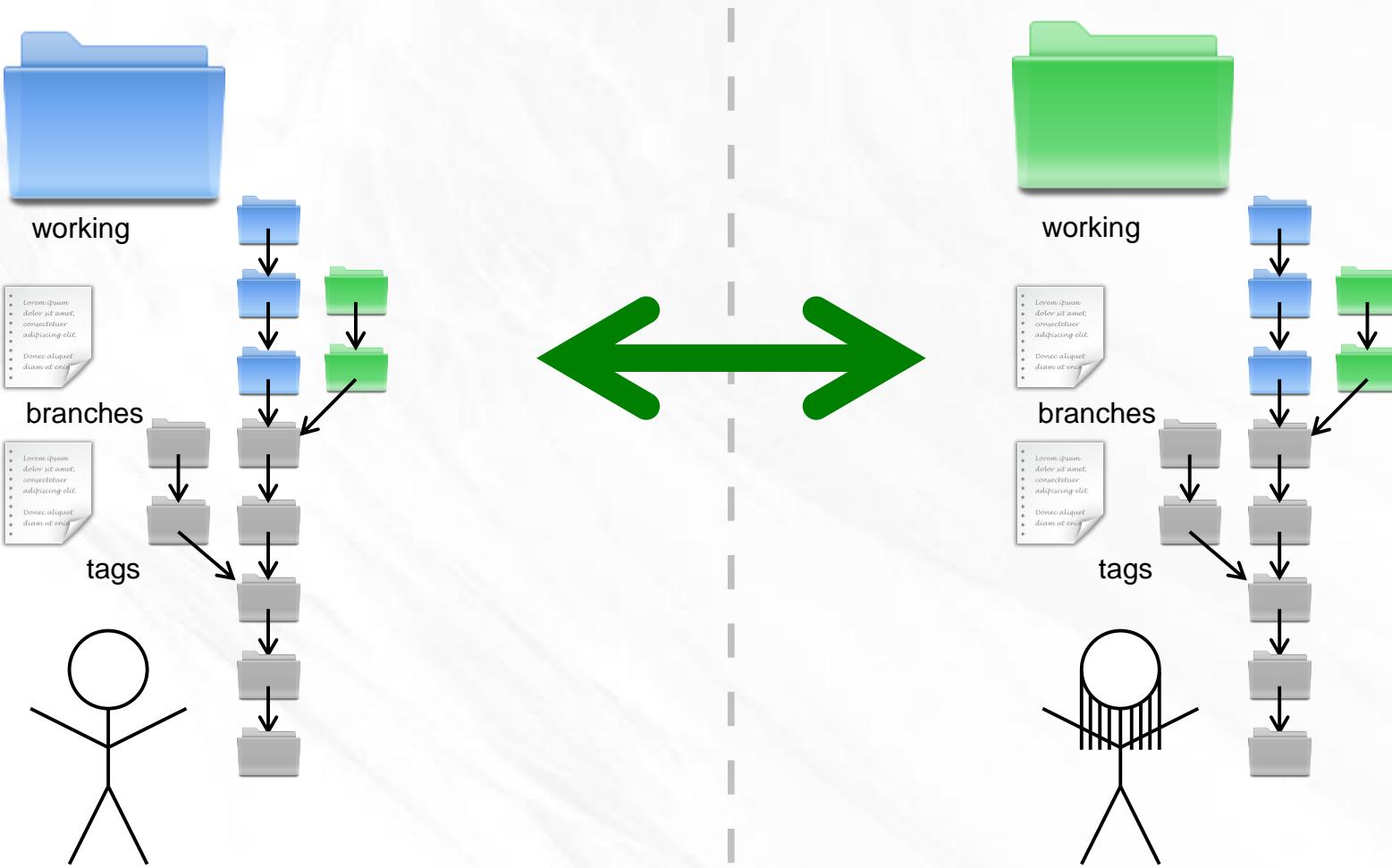
Distributed



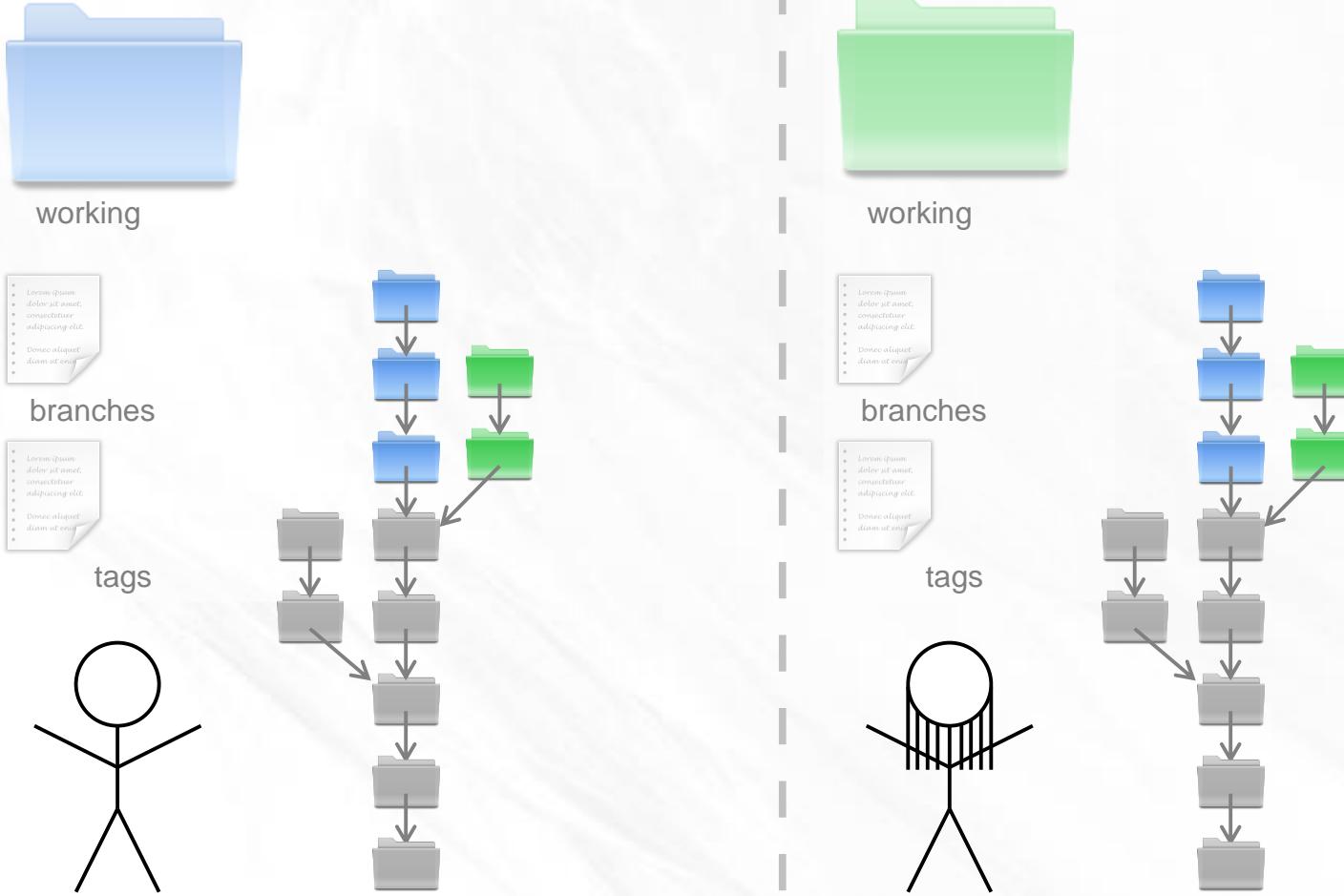
Offline



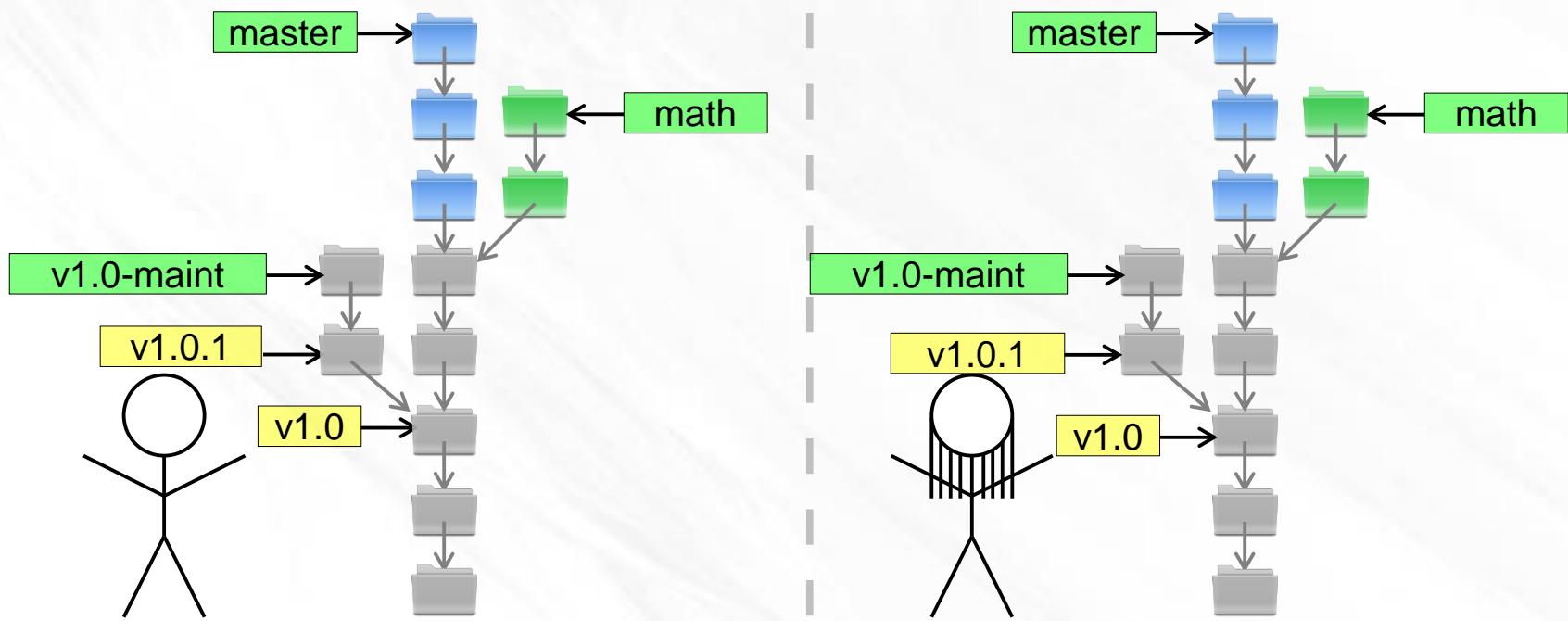
Offline



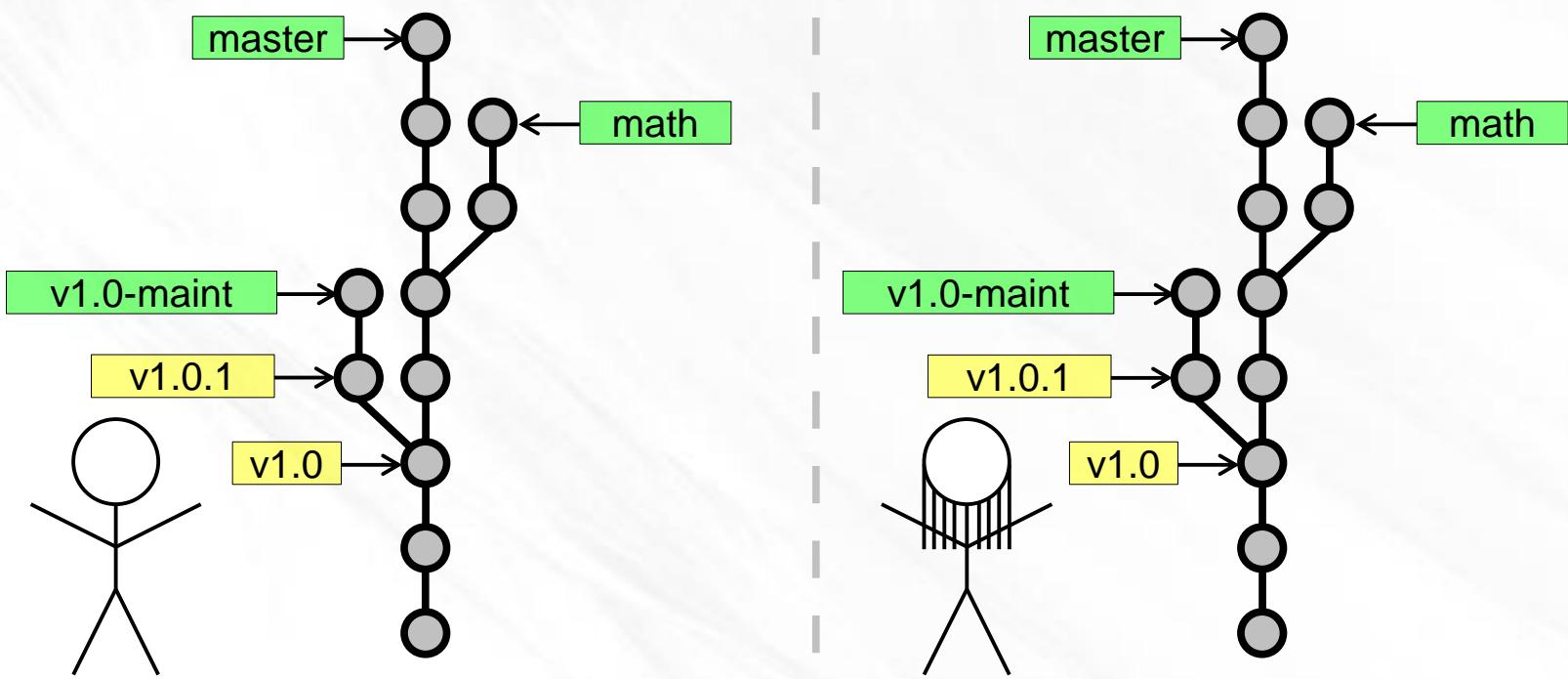
(simpler drawings)



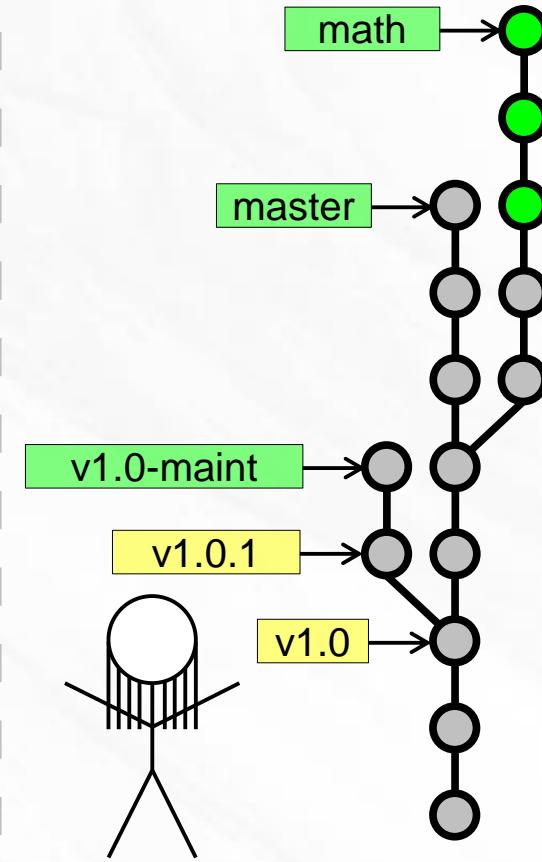
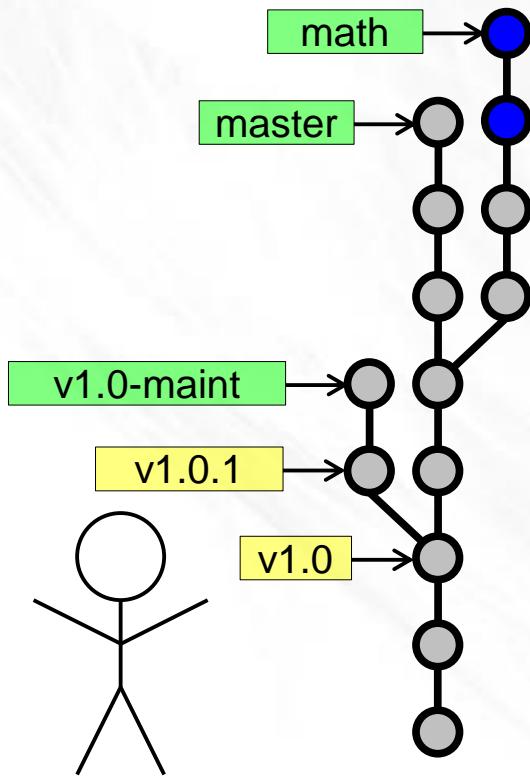
(simpler drawings)



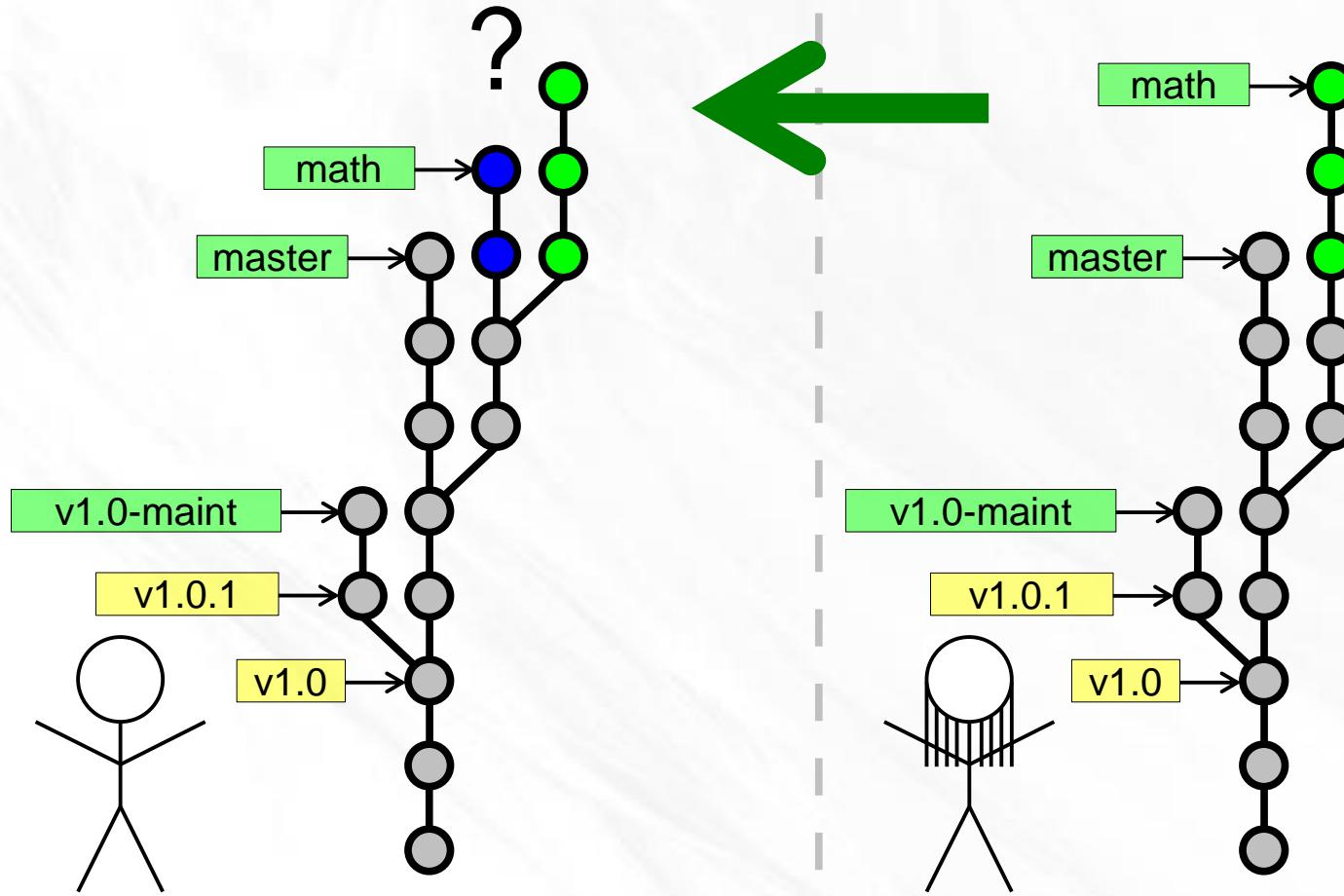
(simpler drawings)



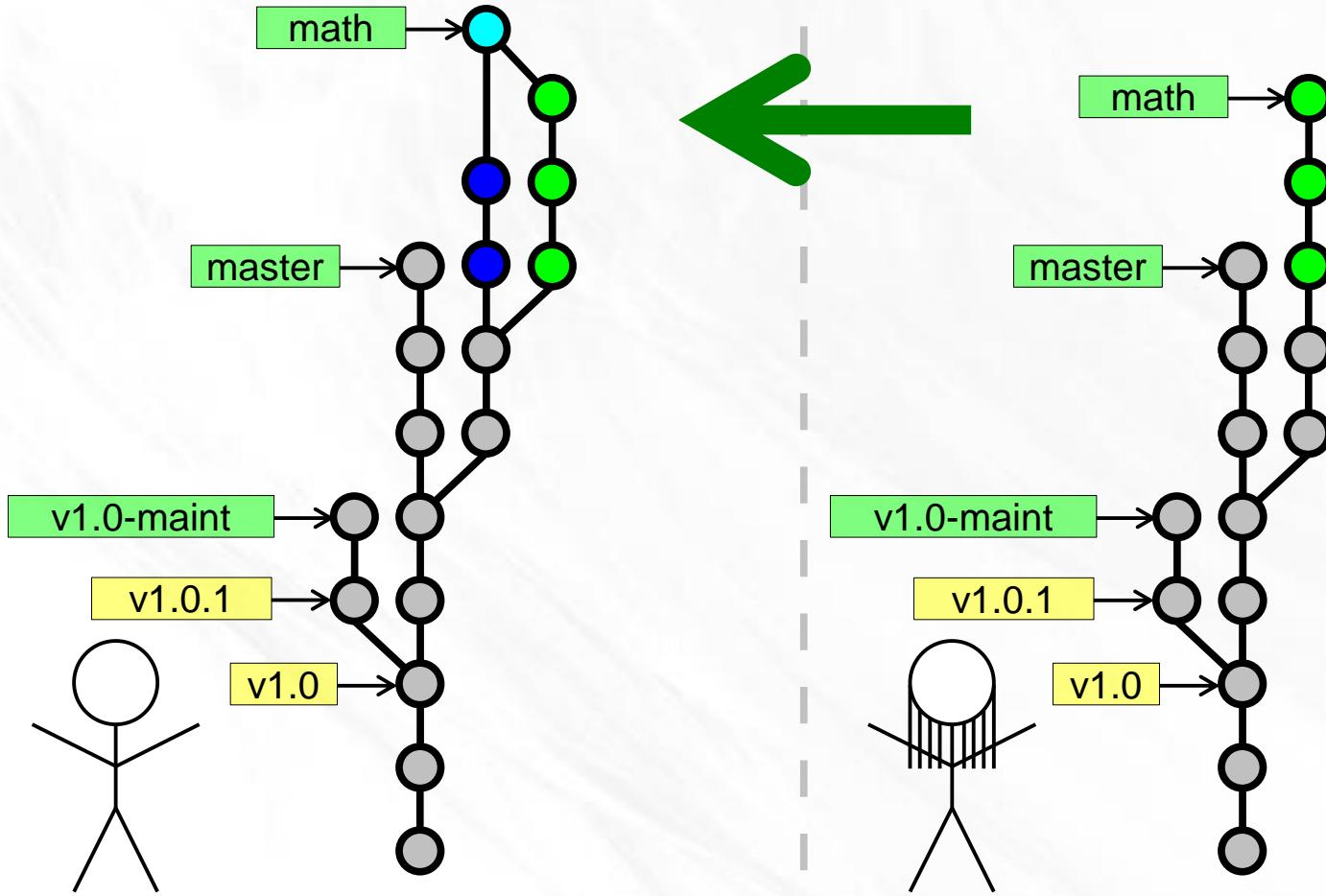
Merges



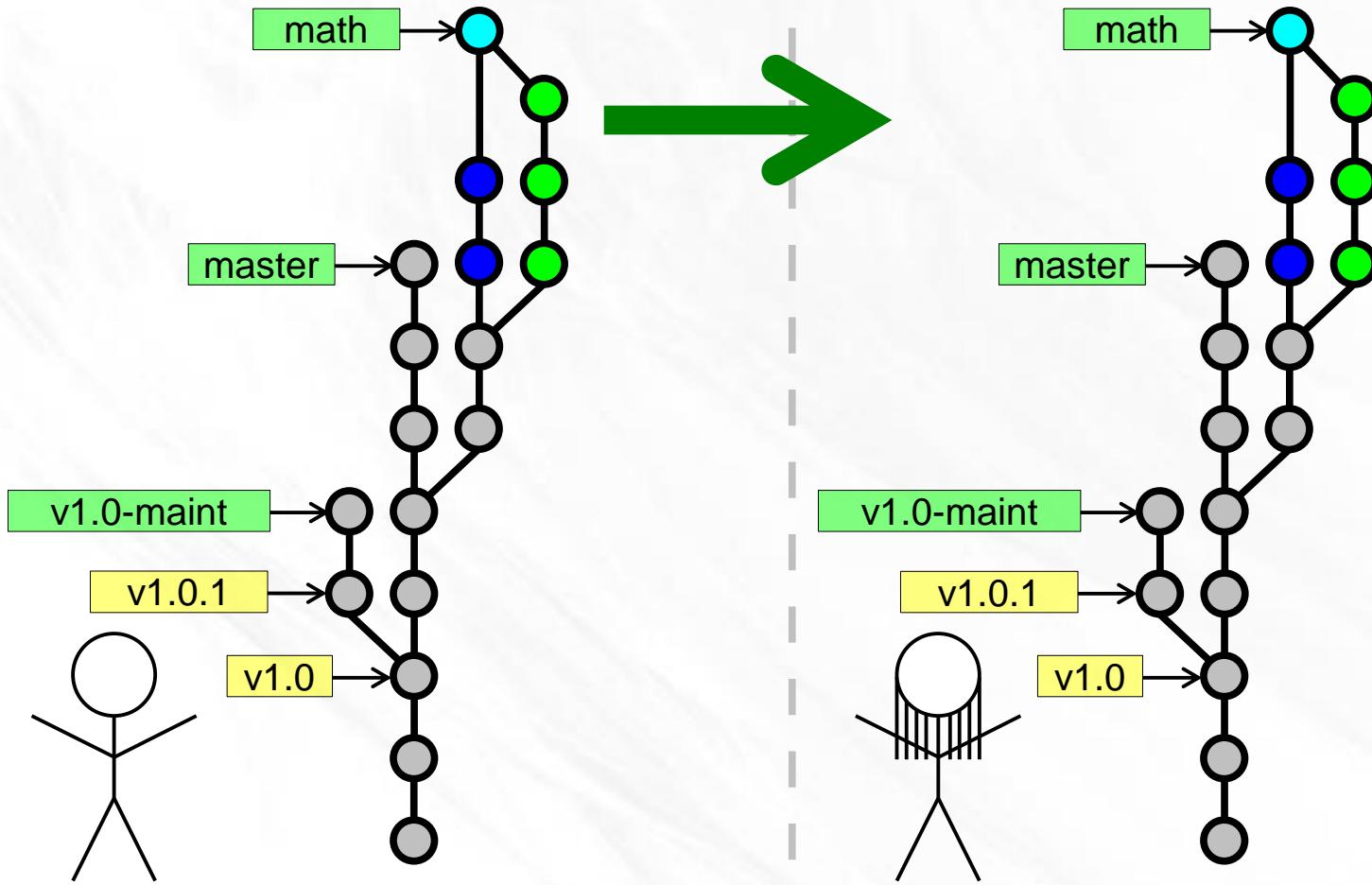
Merges



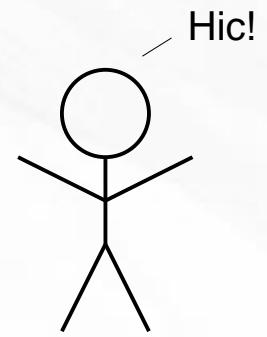
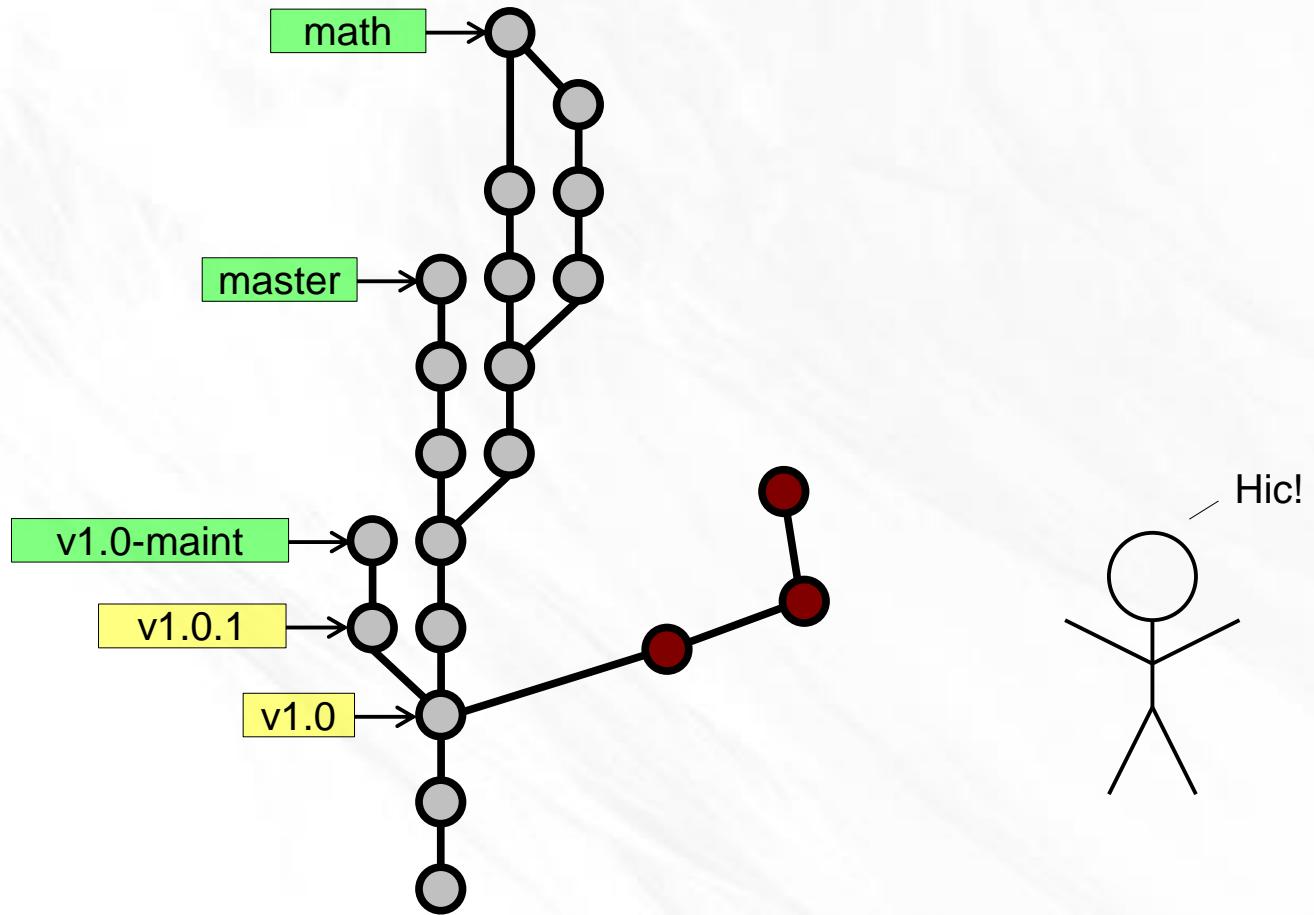
Merges



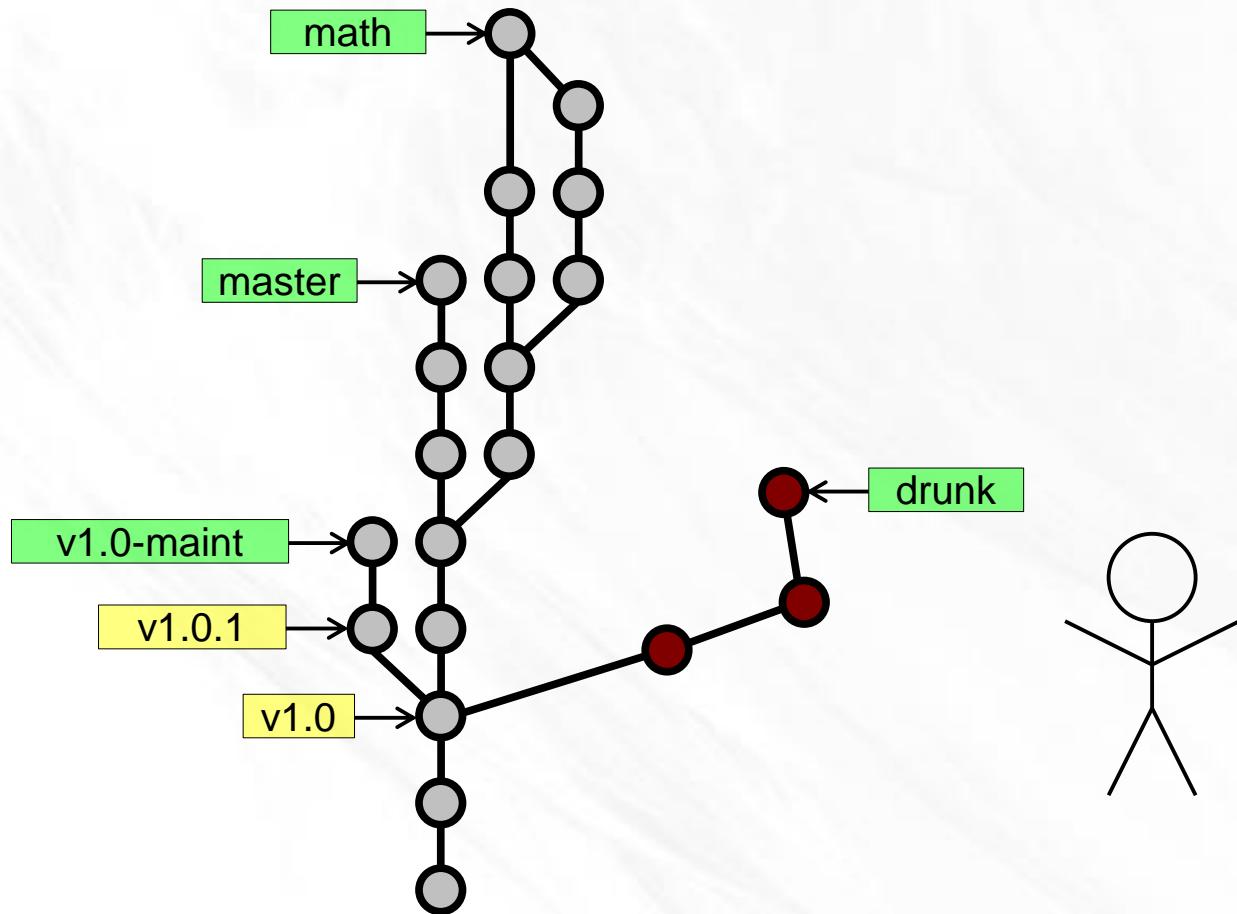
Merges



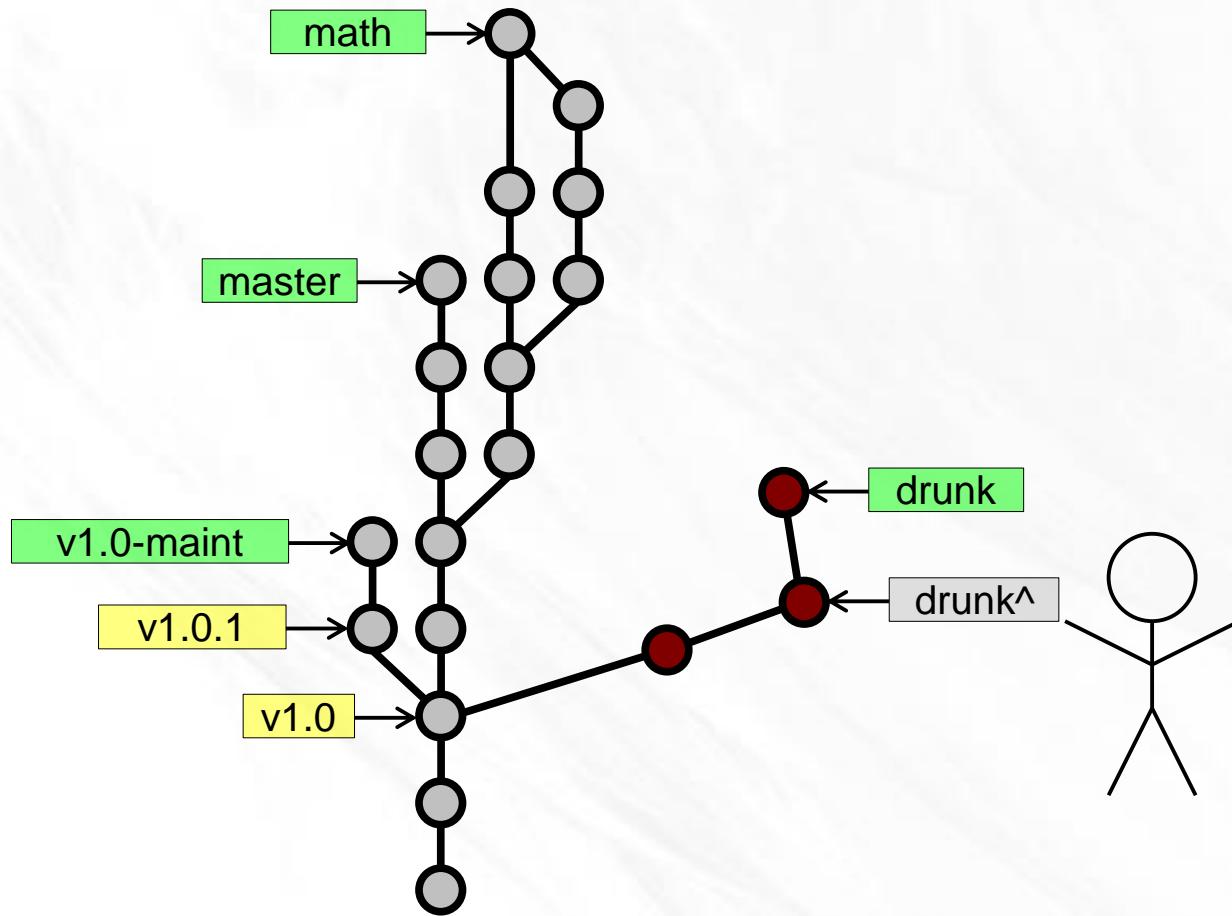
Rewriting History



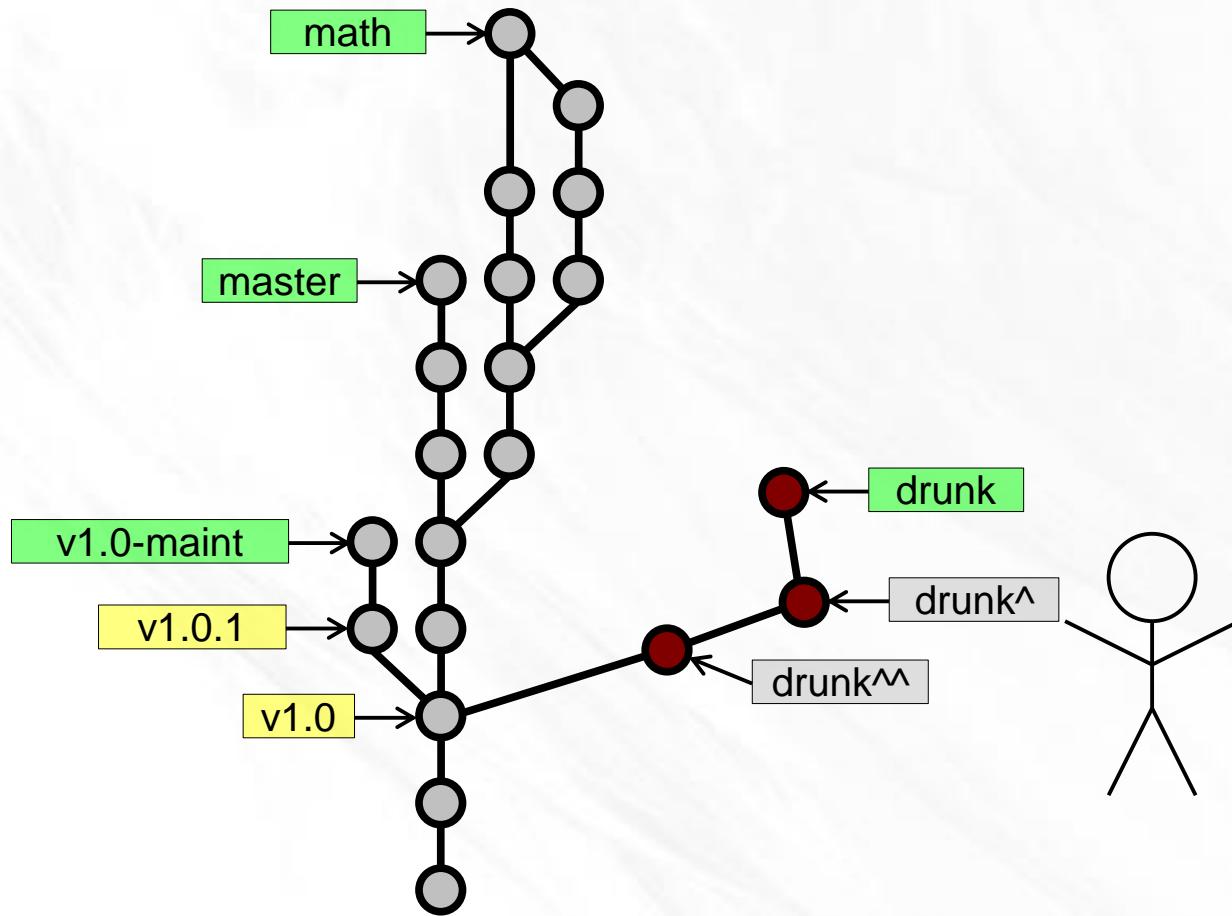
Rewriting History



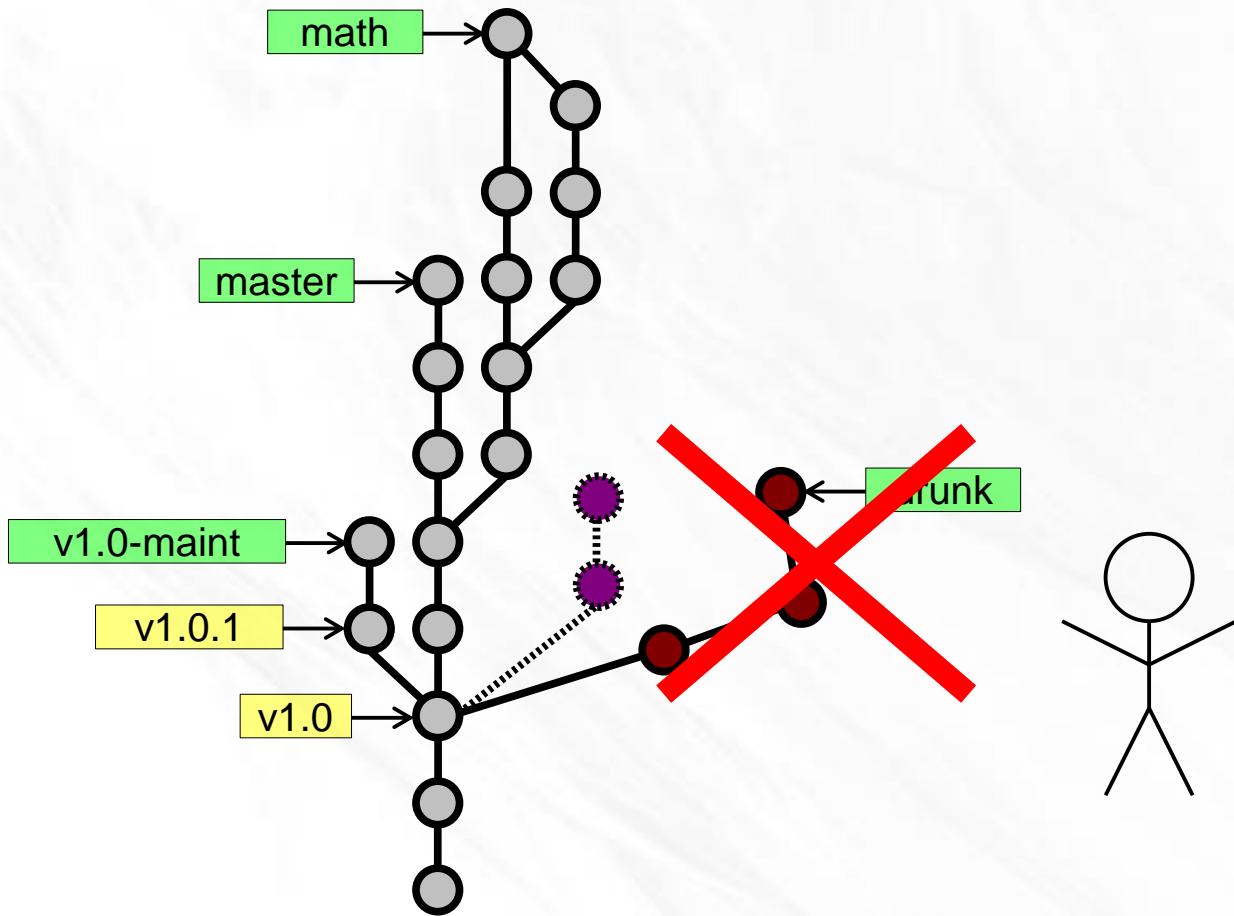
Rewriting History



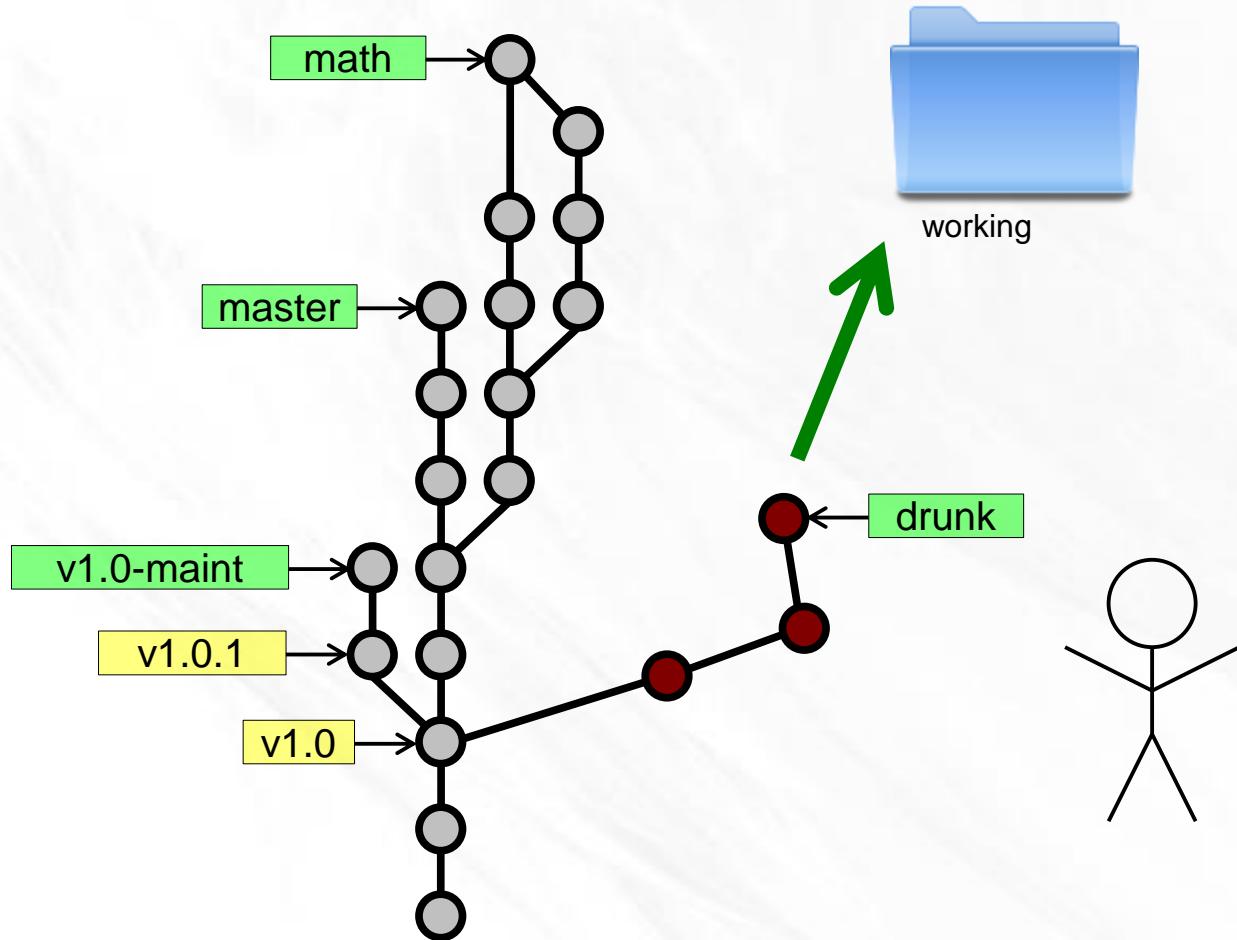
Rewriting History



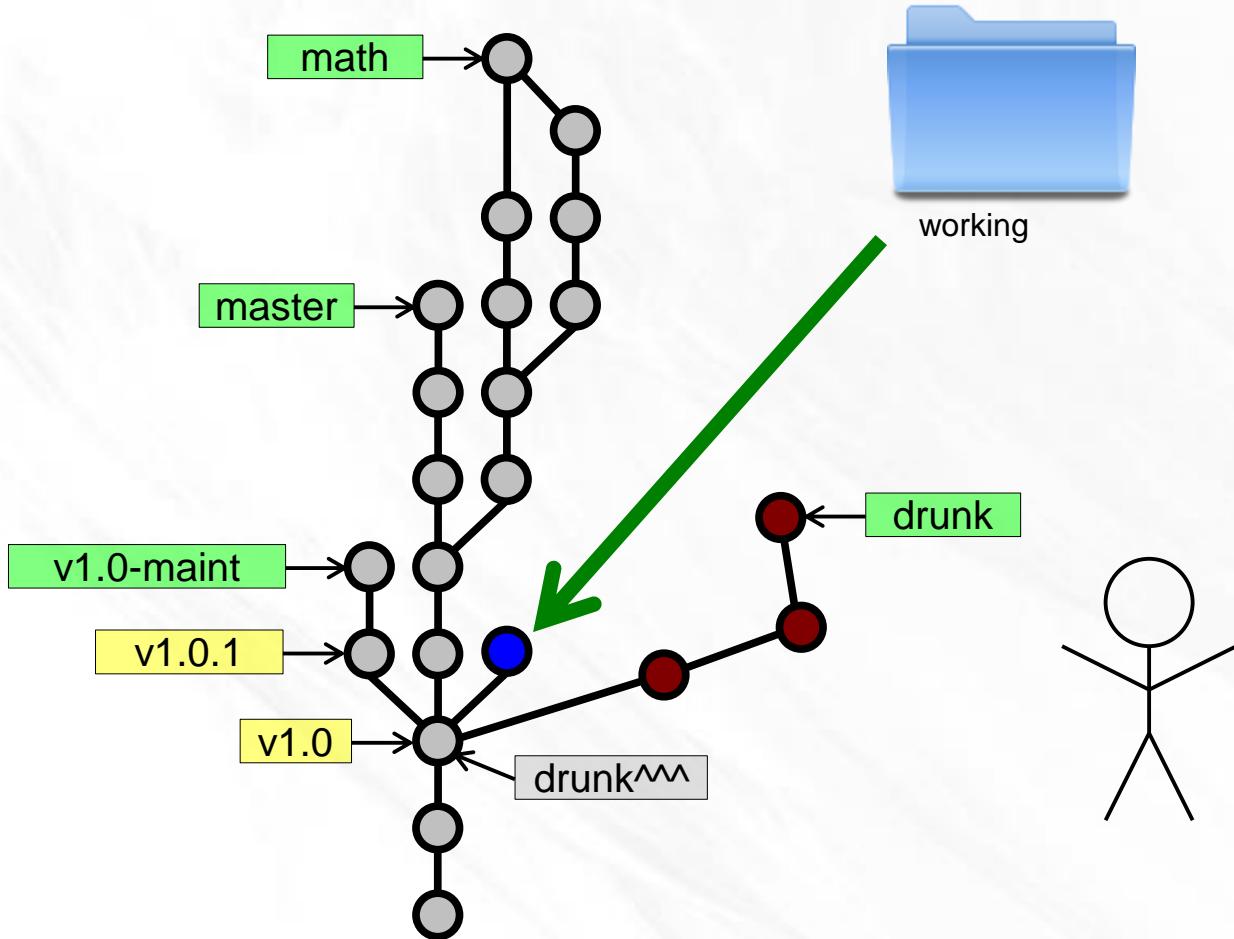
Rewriting History



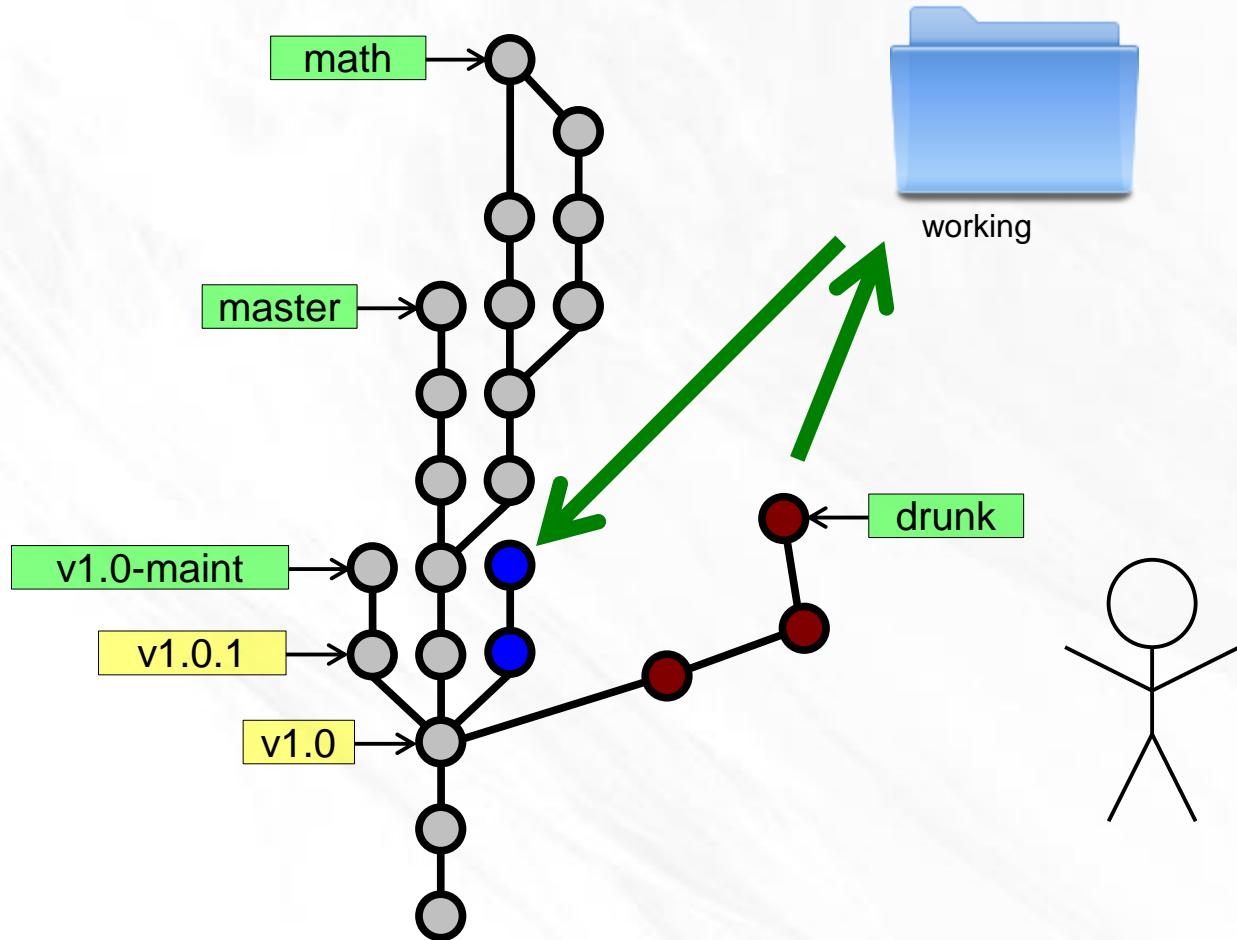
Rewriting History



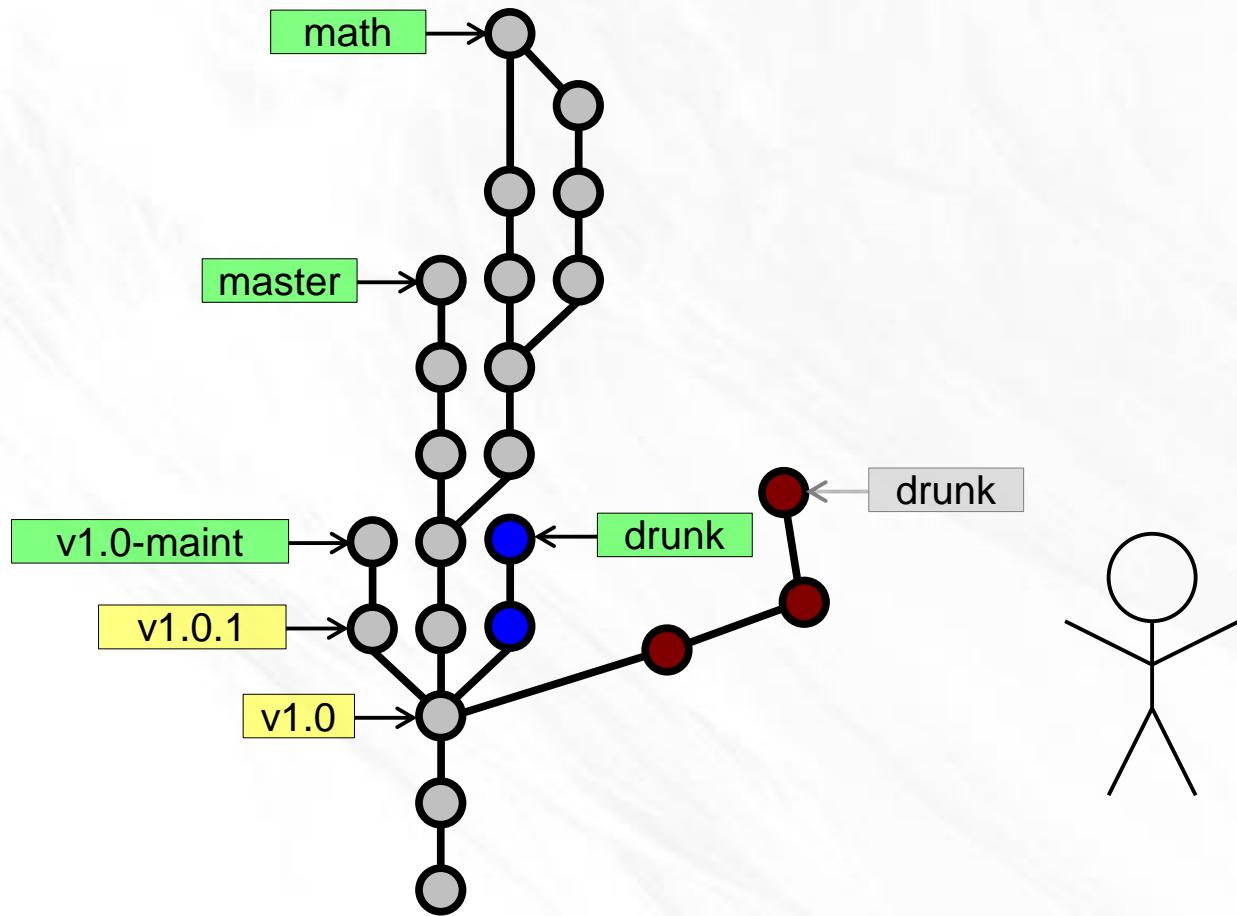
Rewriting History



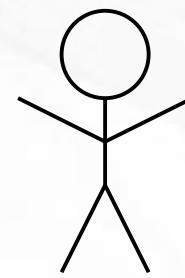
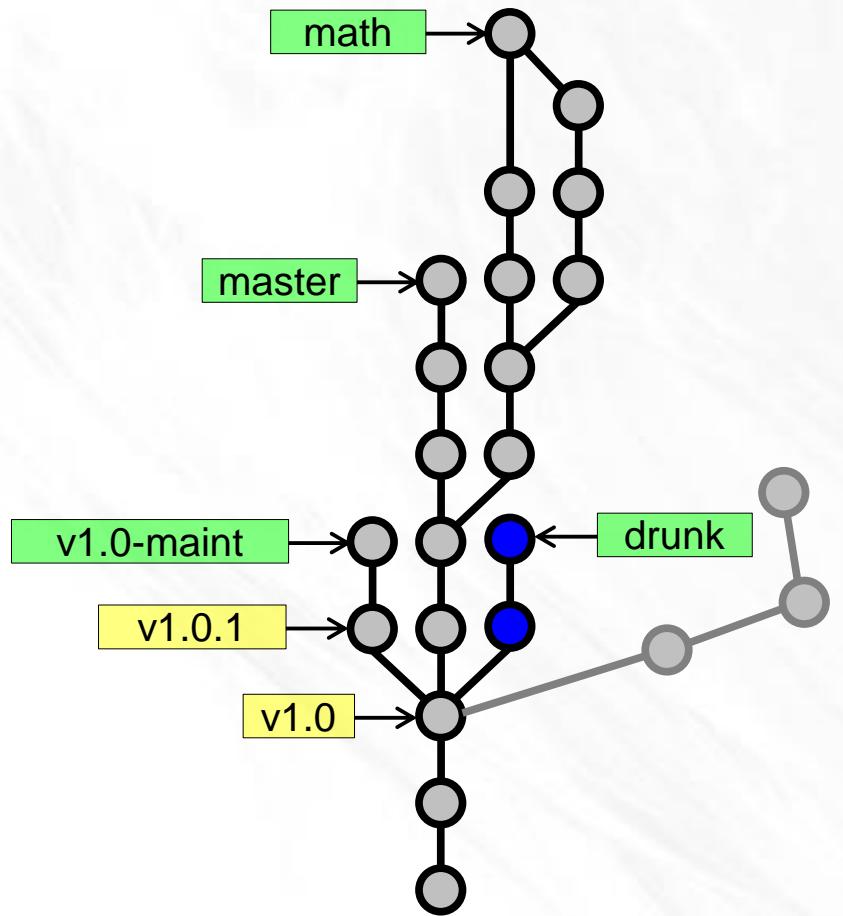
Rewriting History



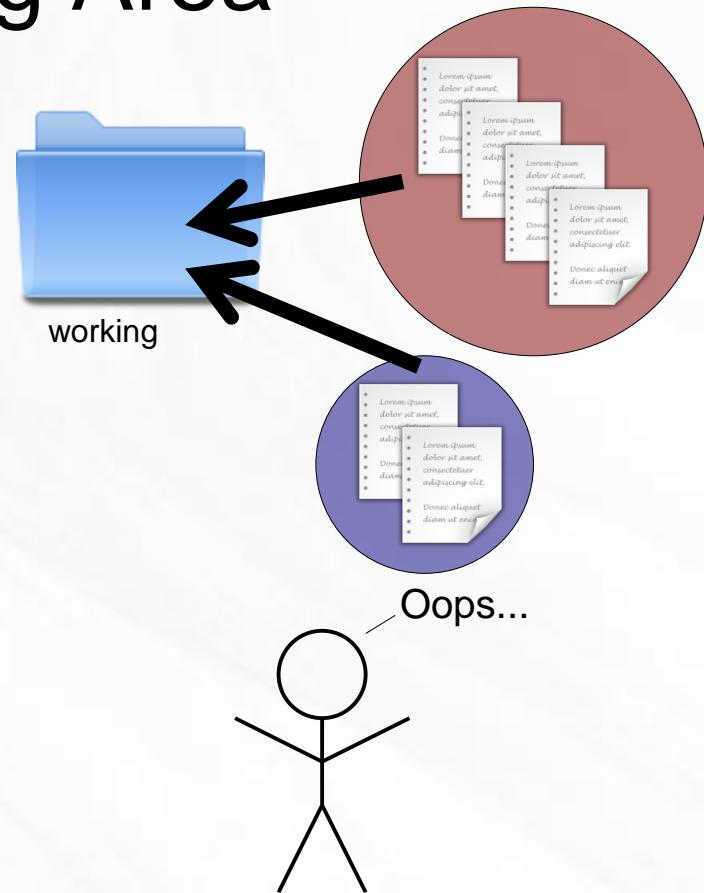
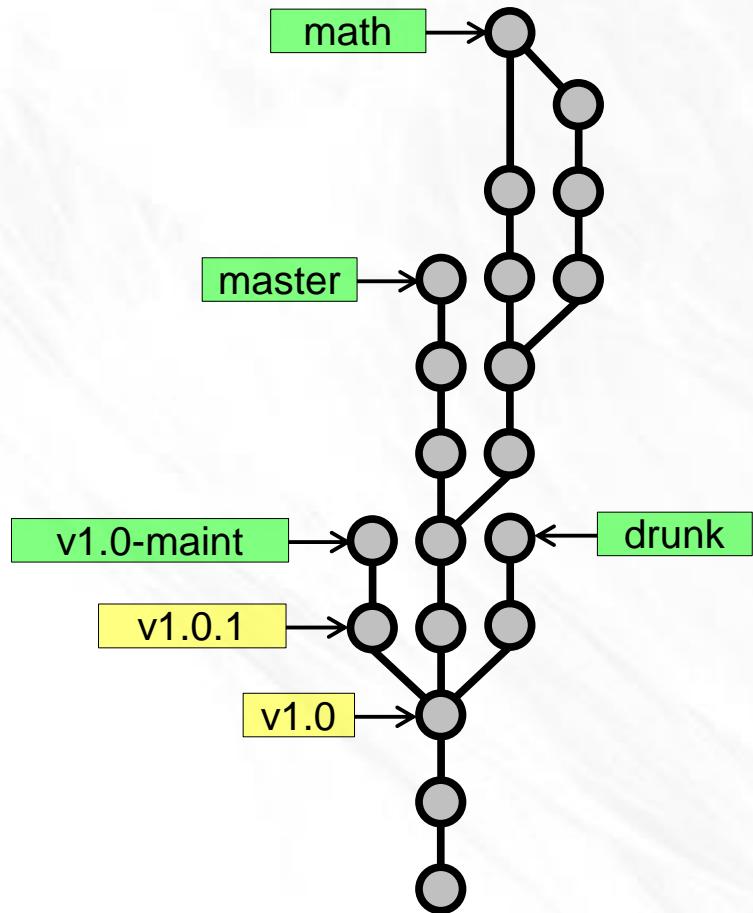
Rewriting History



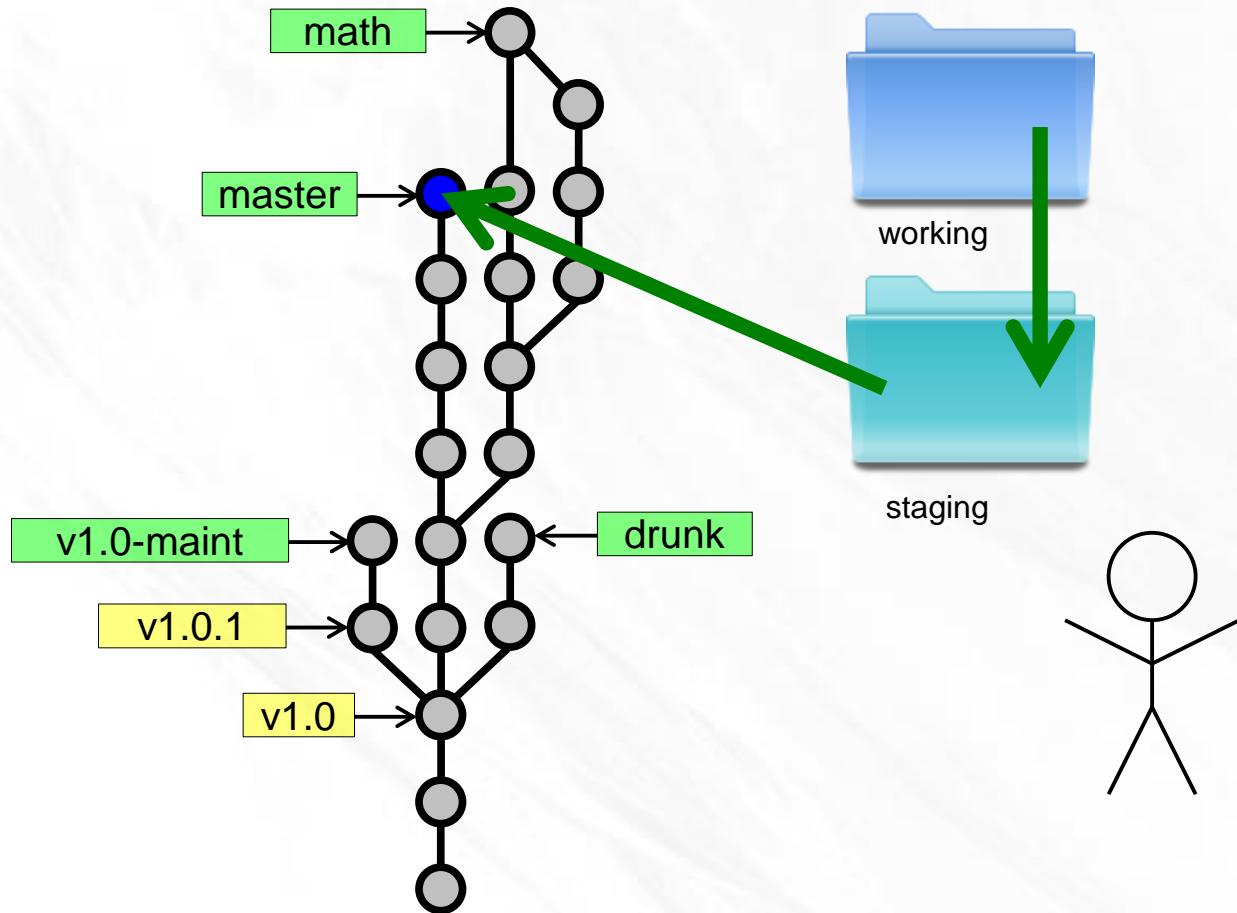
Rewriting History



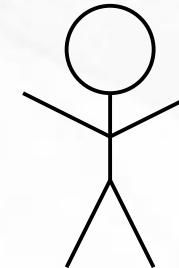
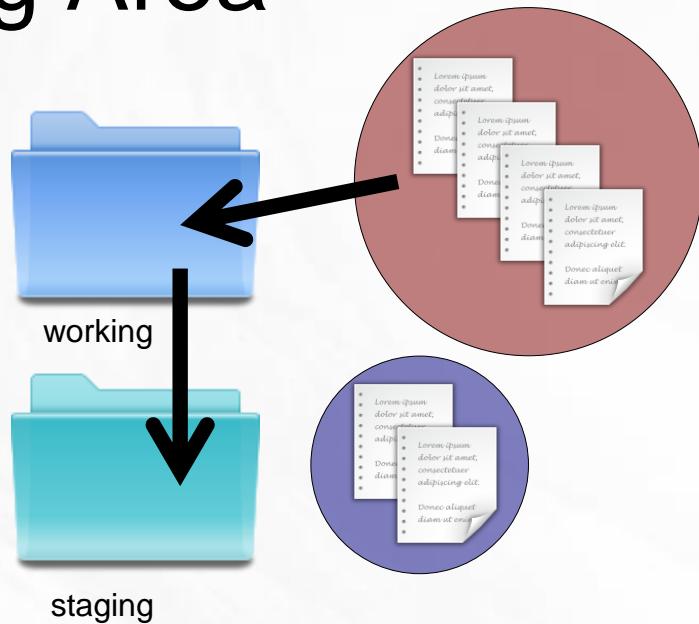
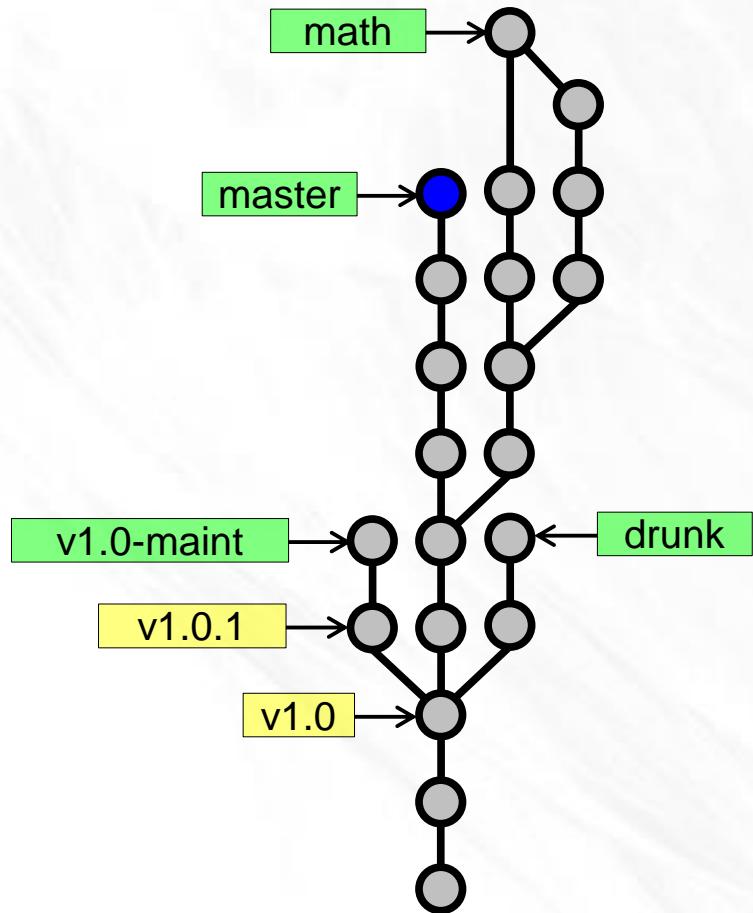
Staging Area



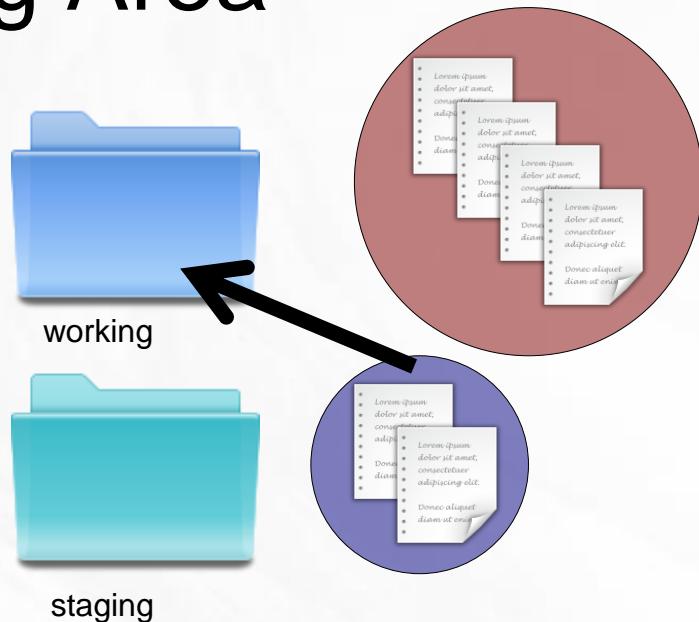
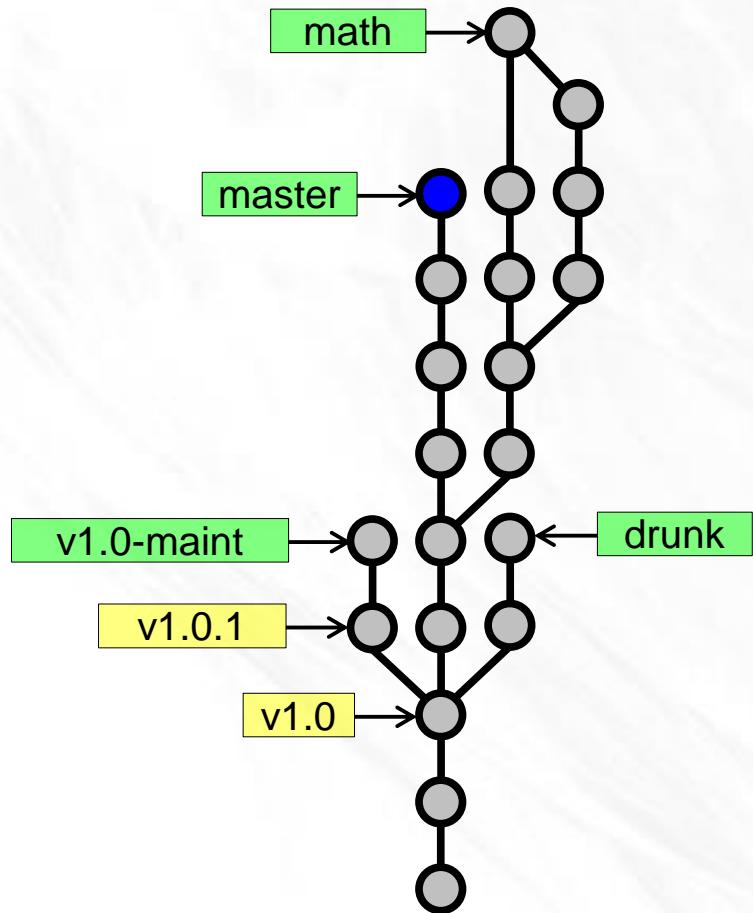
Staging Area



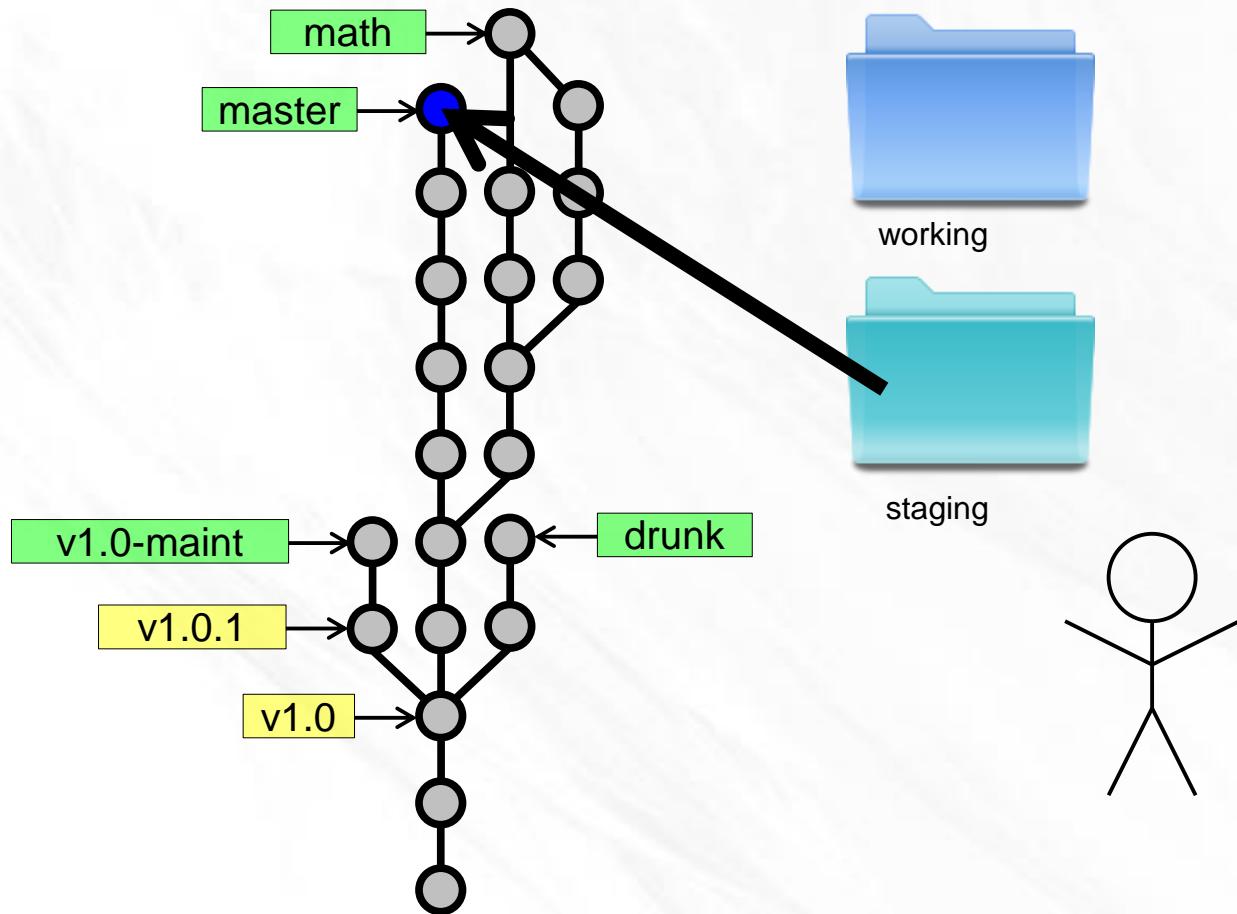
Staging Area



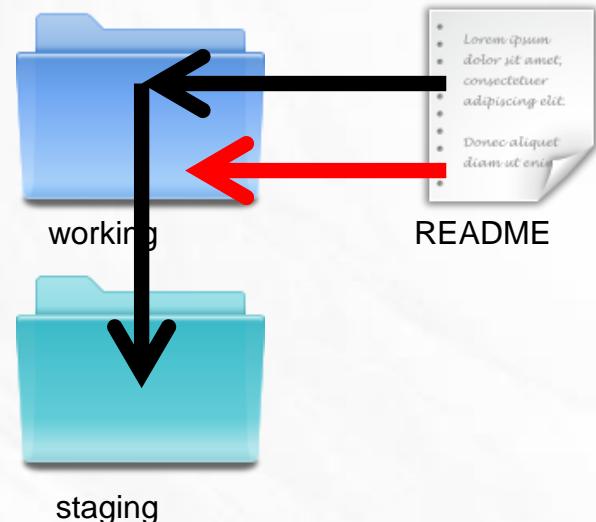
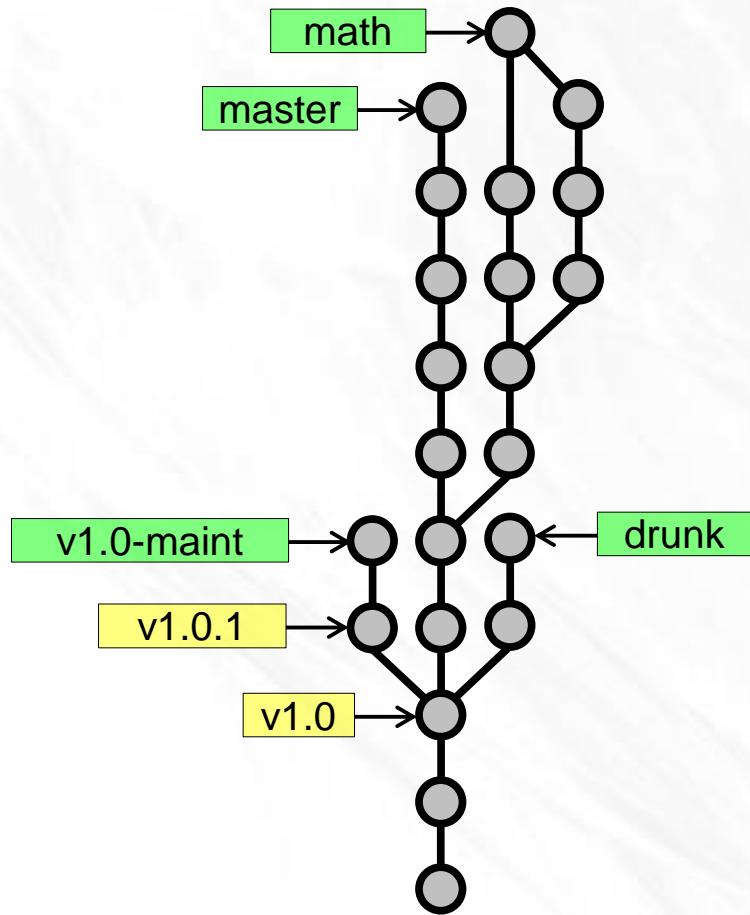
Staging Area



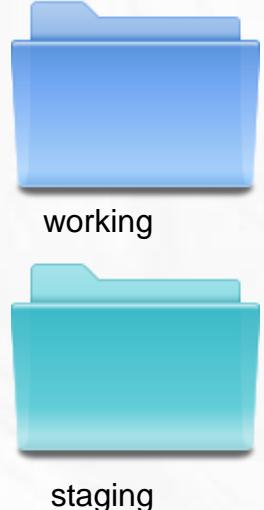
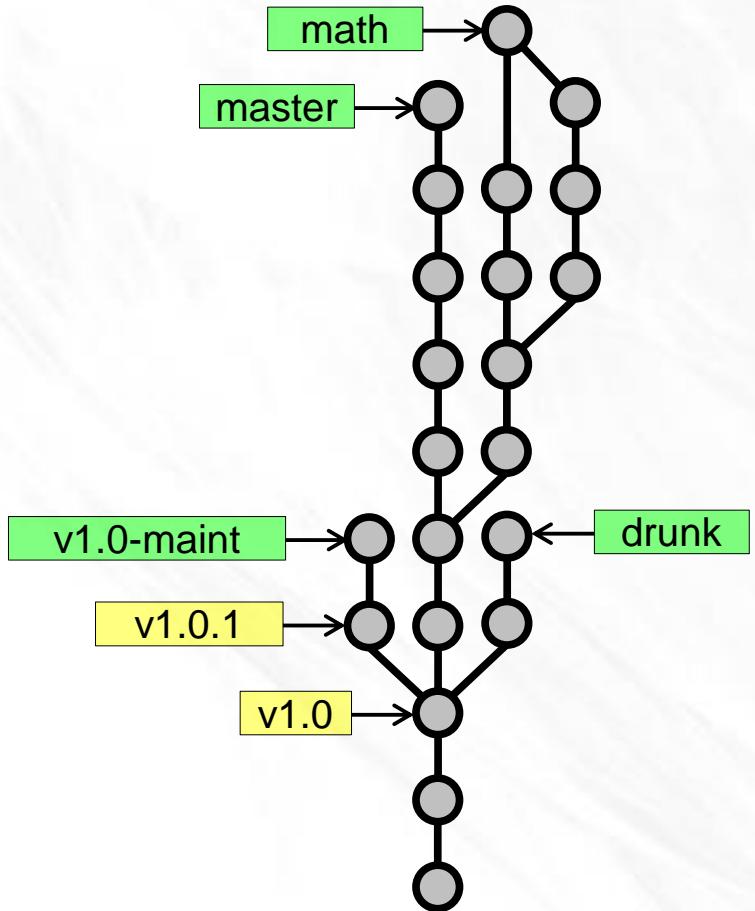
Staging Area



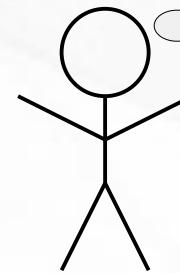
Staging Area



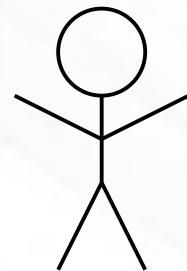
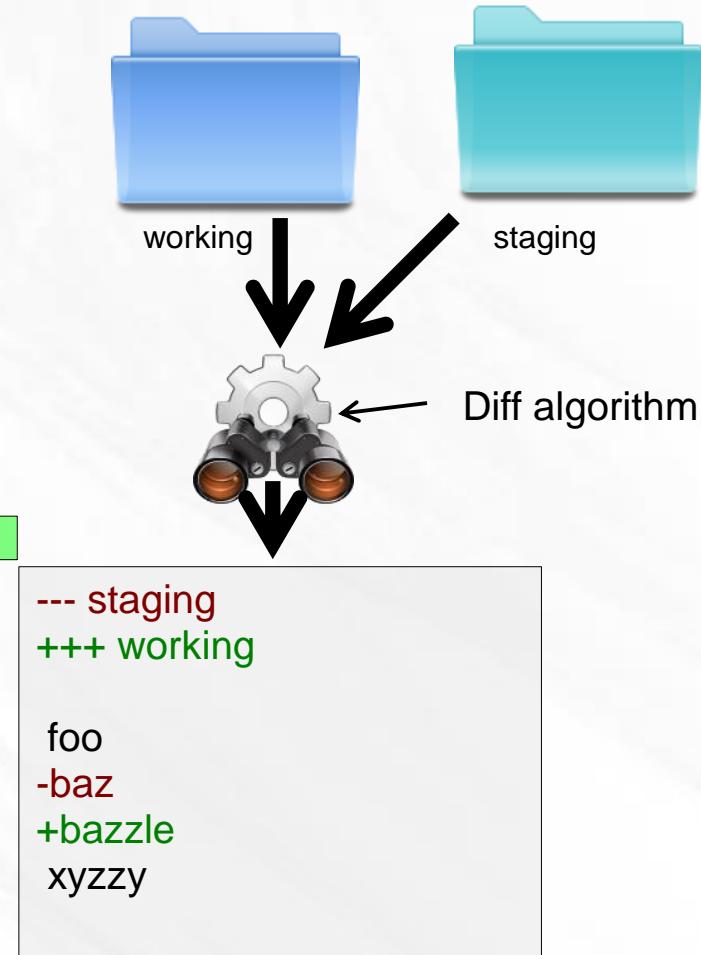
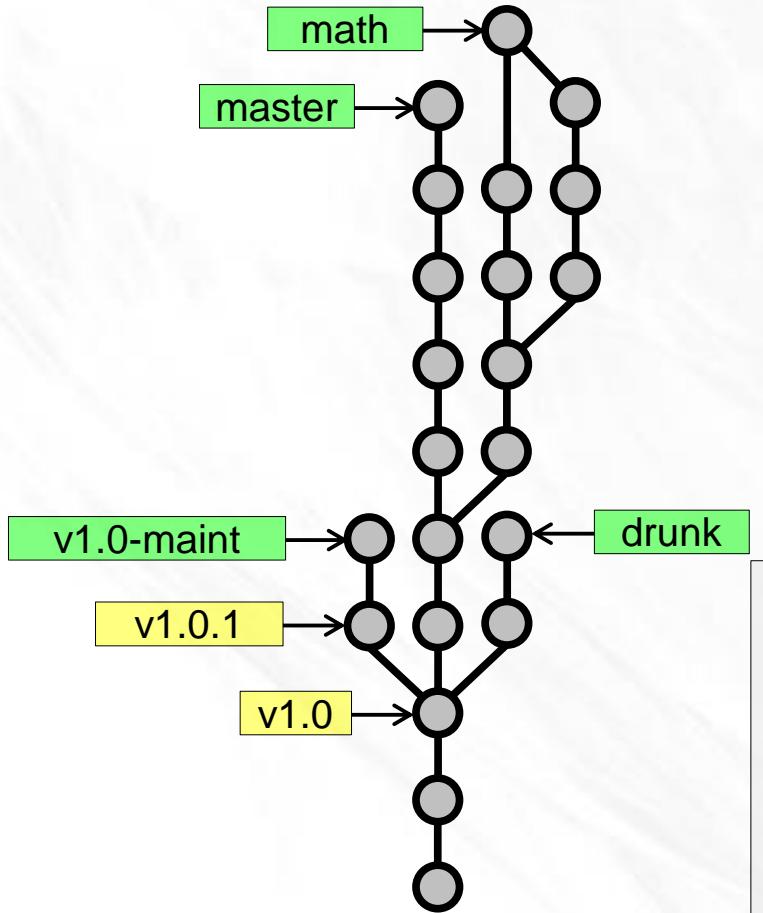
Diffs



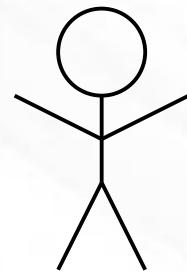
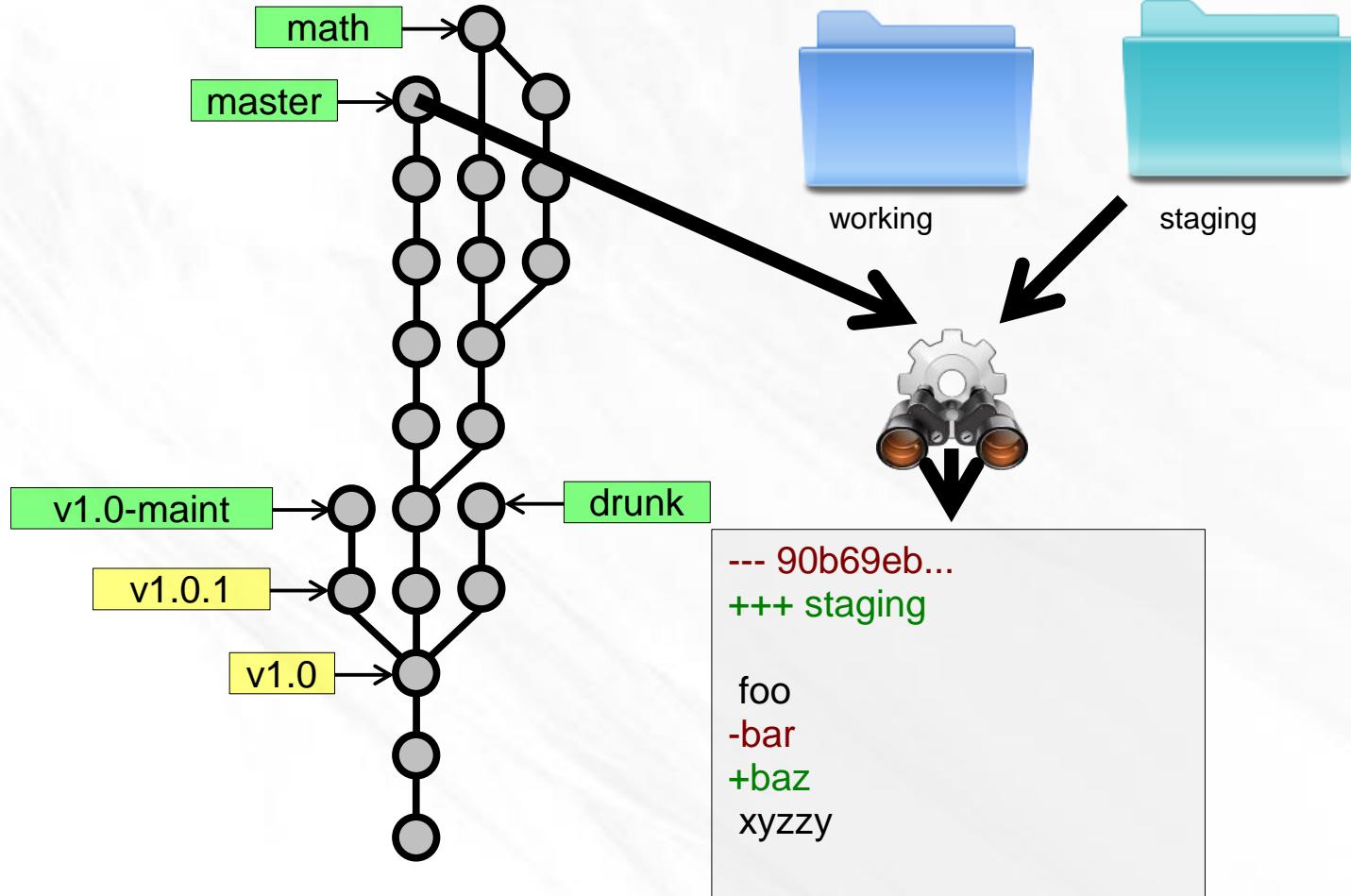
What are the changes?
- working vs. staging
- working vs. snapshot X
- staging vs. snapshot X
- snapshot X vs. snapshot Y



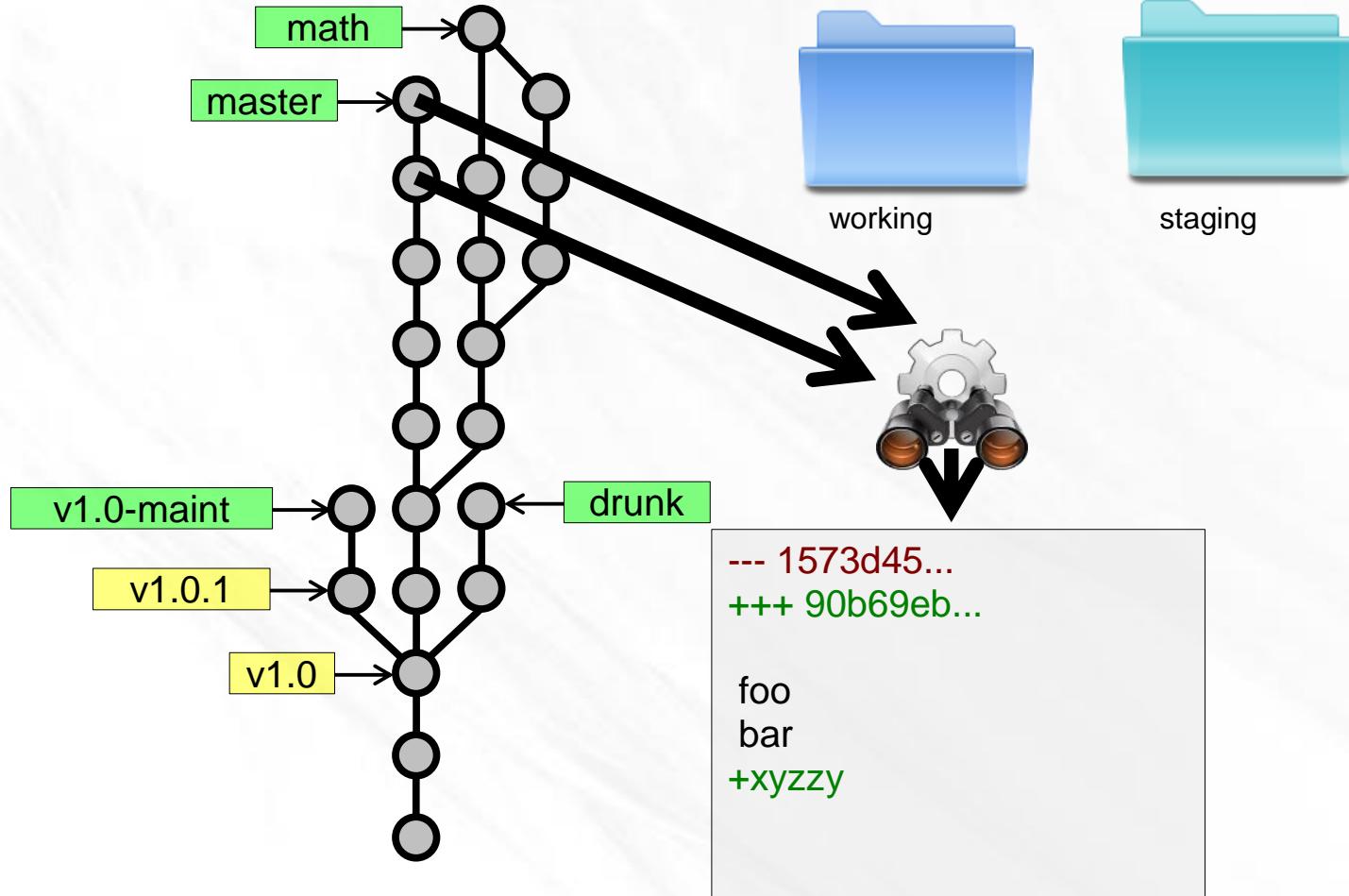
Diffs



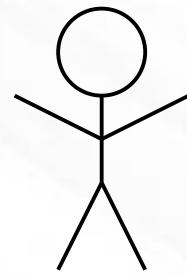
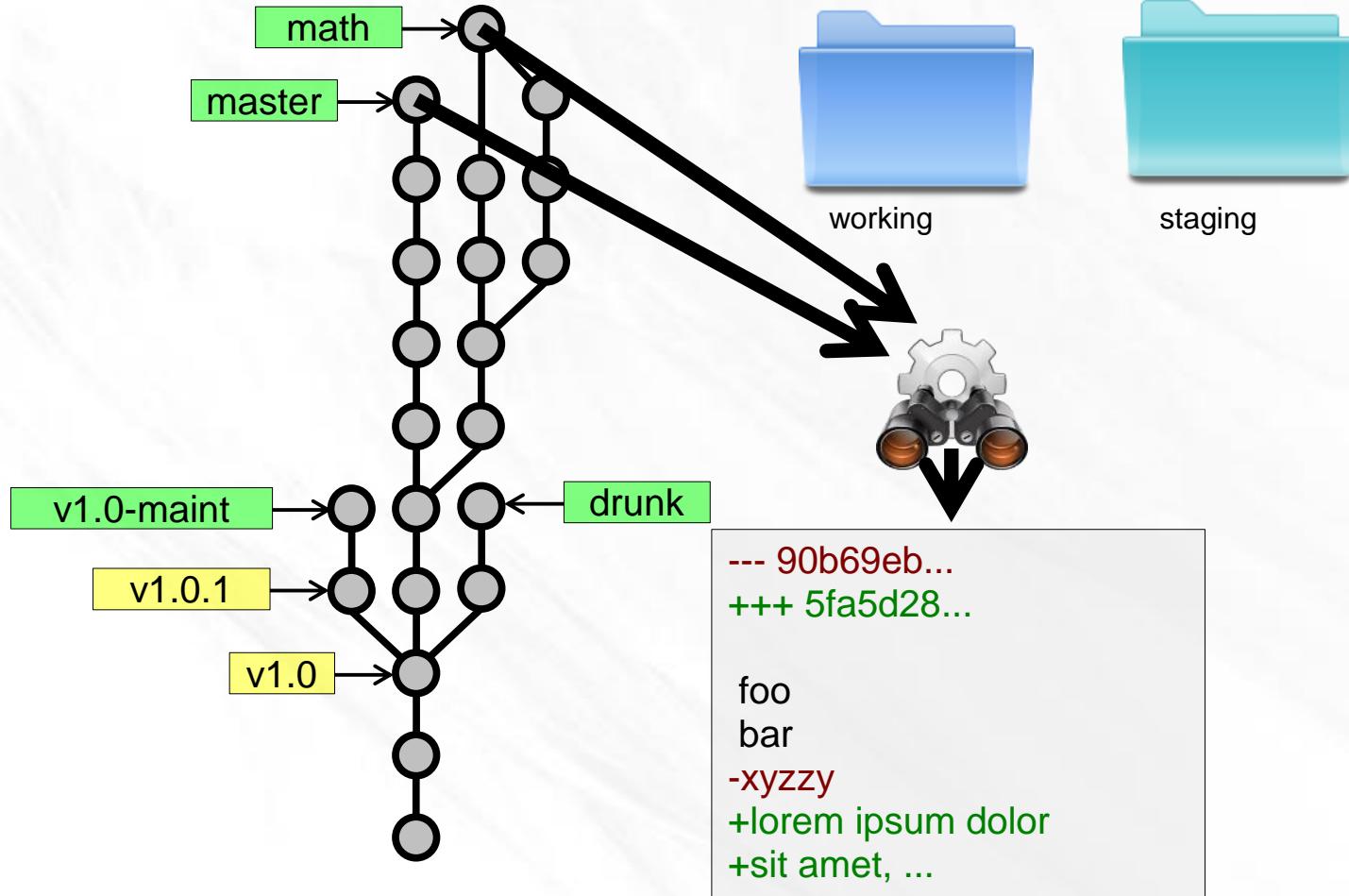
Diffs



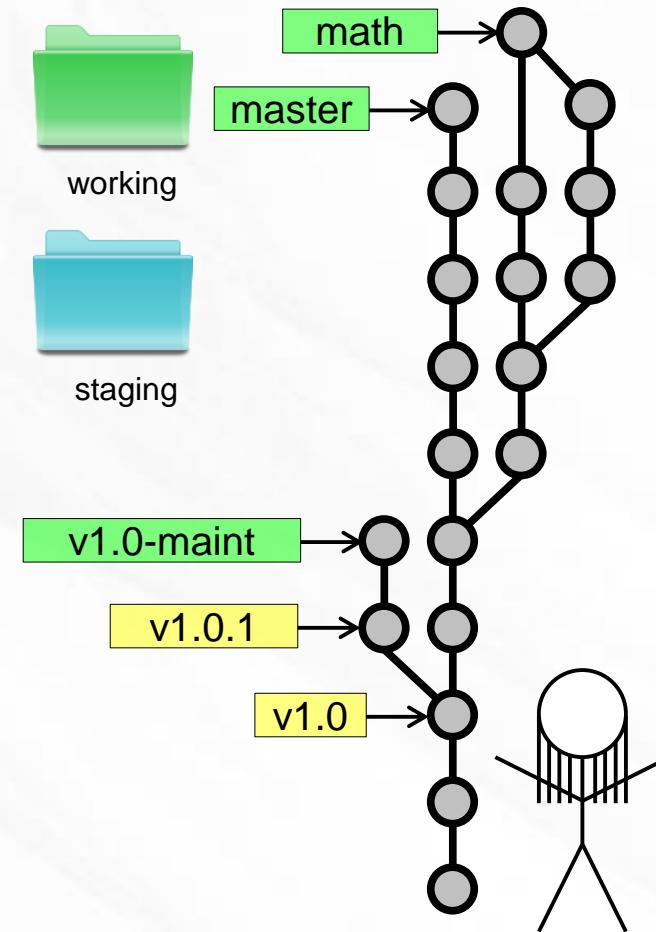
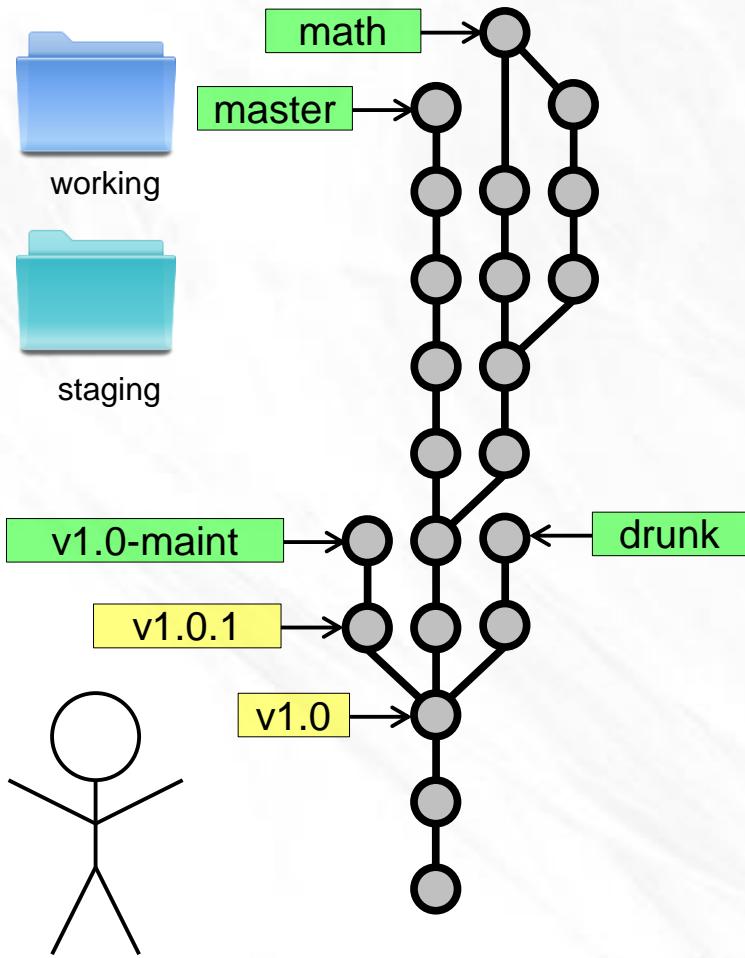
Diffs



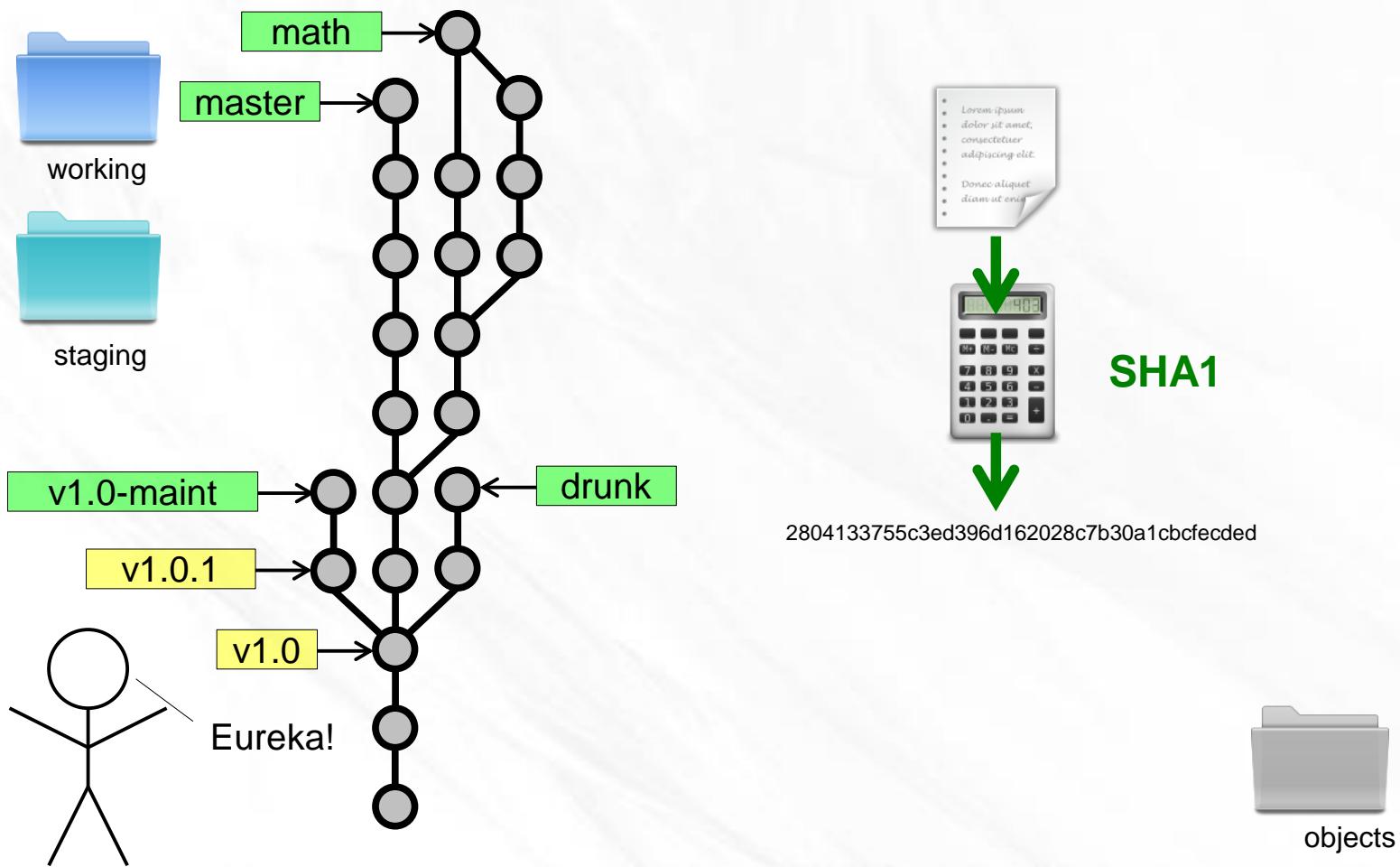
Diffs



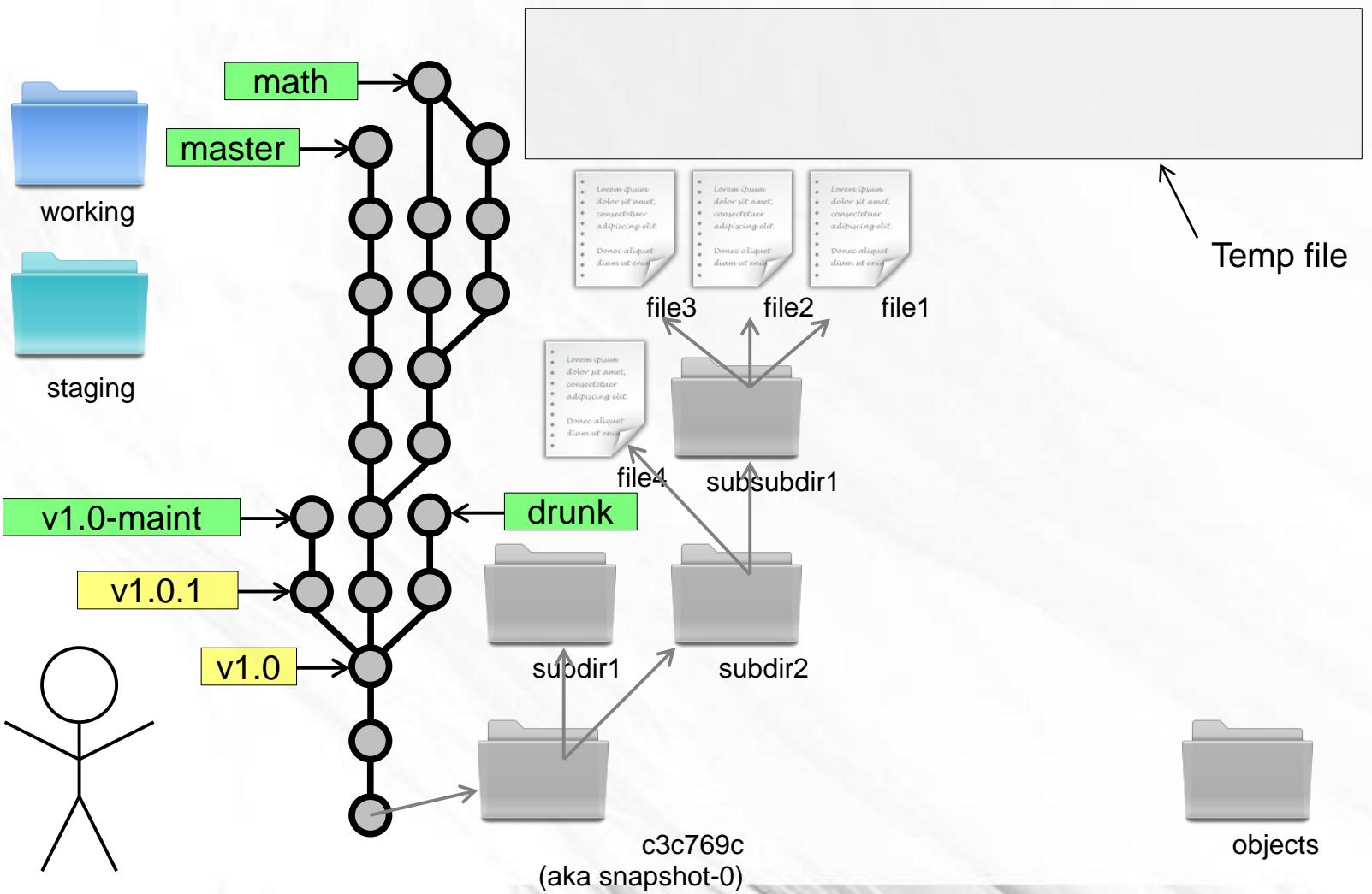
Eliminating Duplication



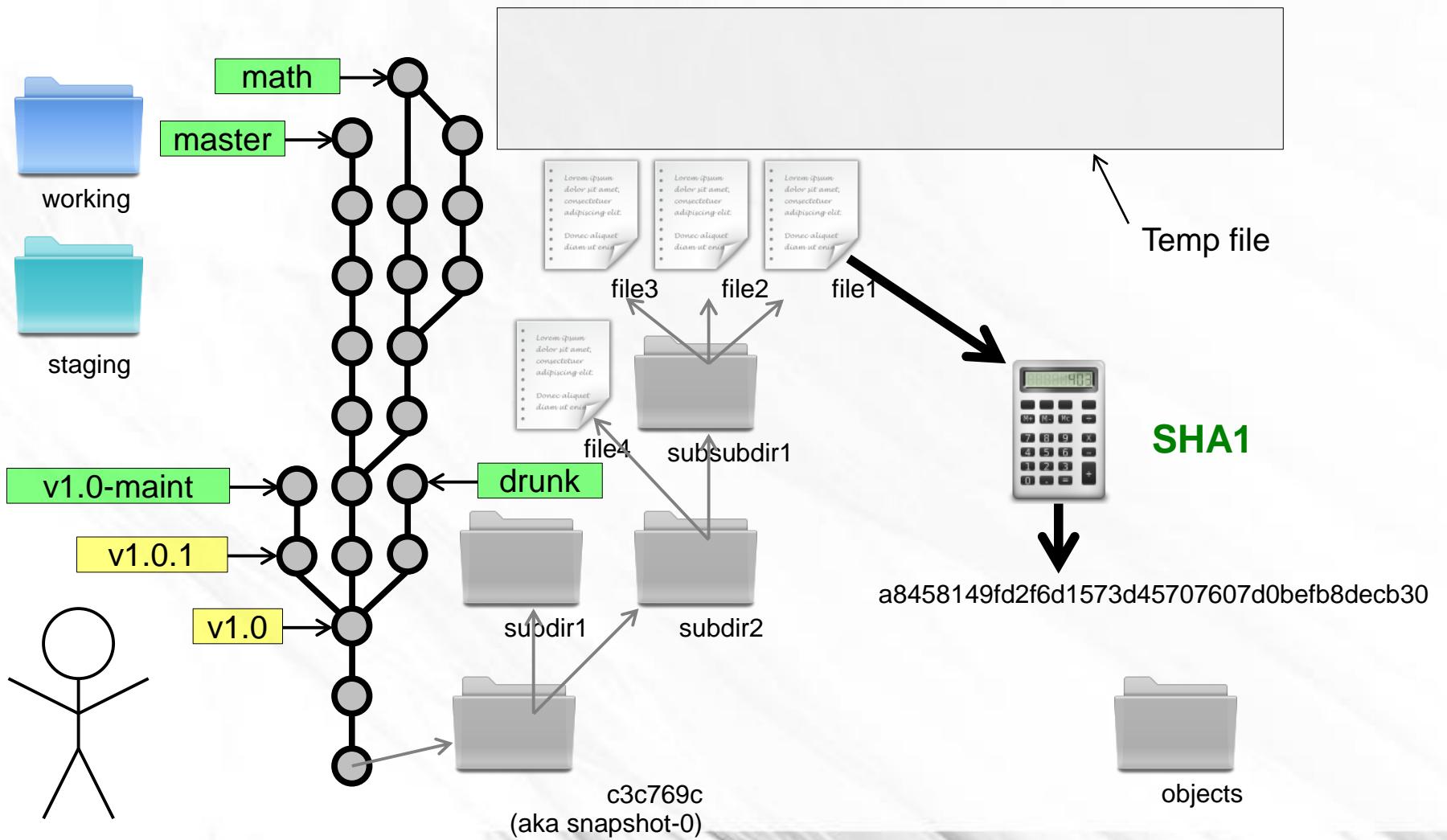
Eliminating Duplication



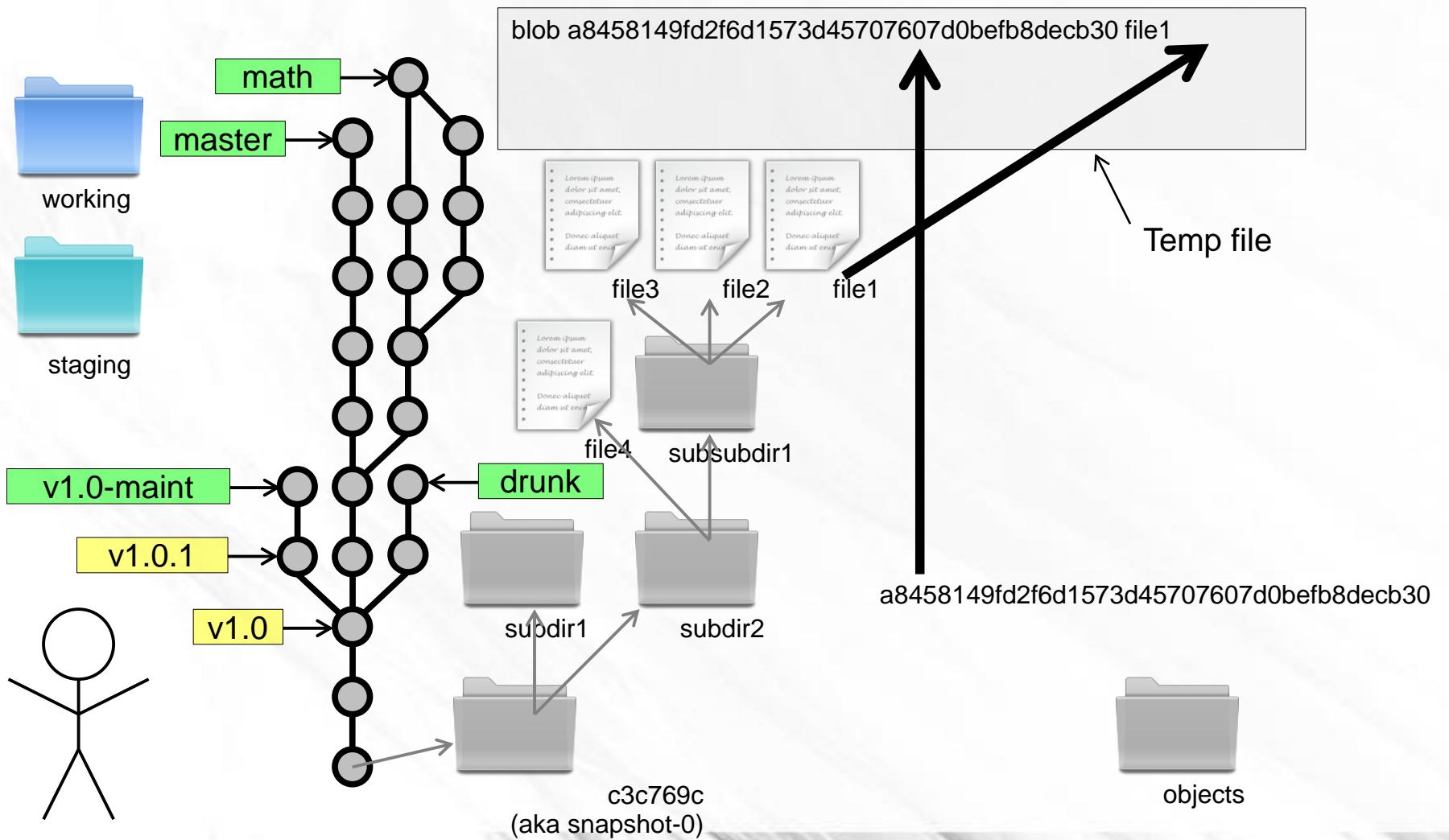
Eliminating Duplication



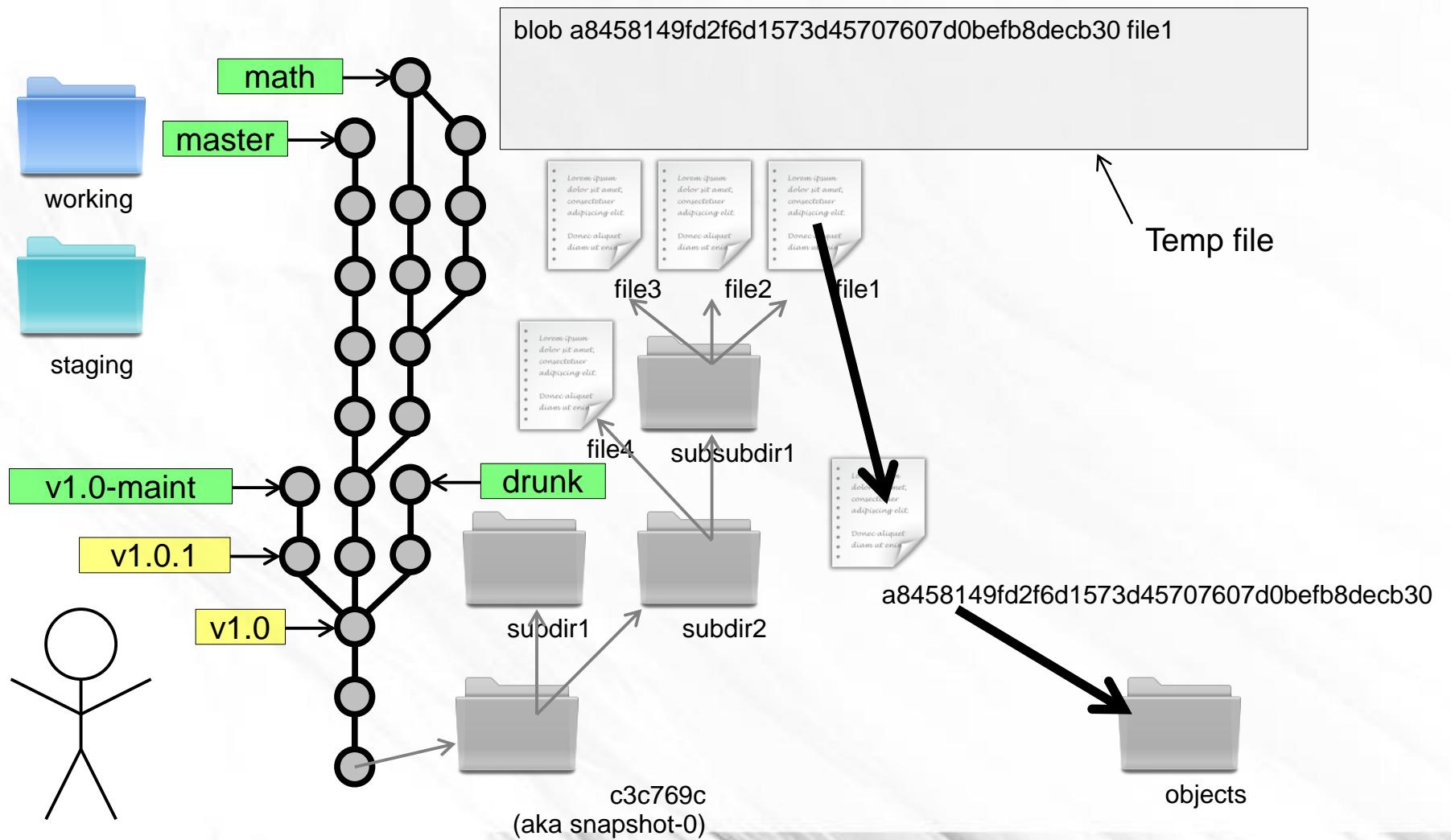
Eliminating Duplication



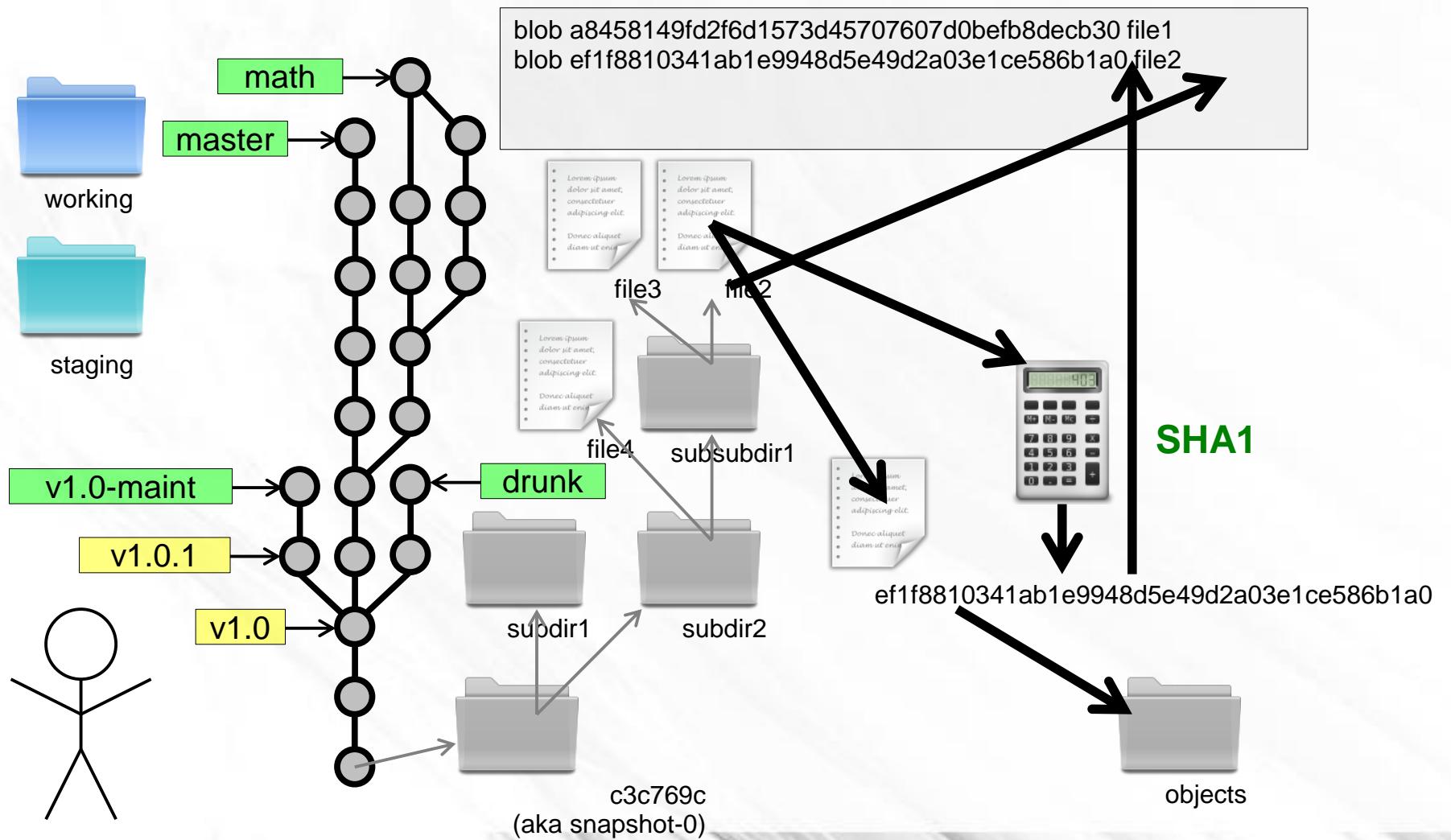
Eliminating Duplication



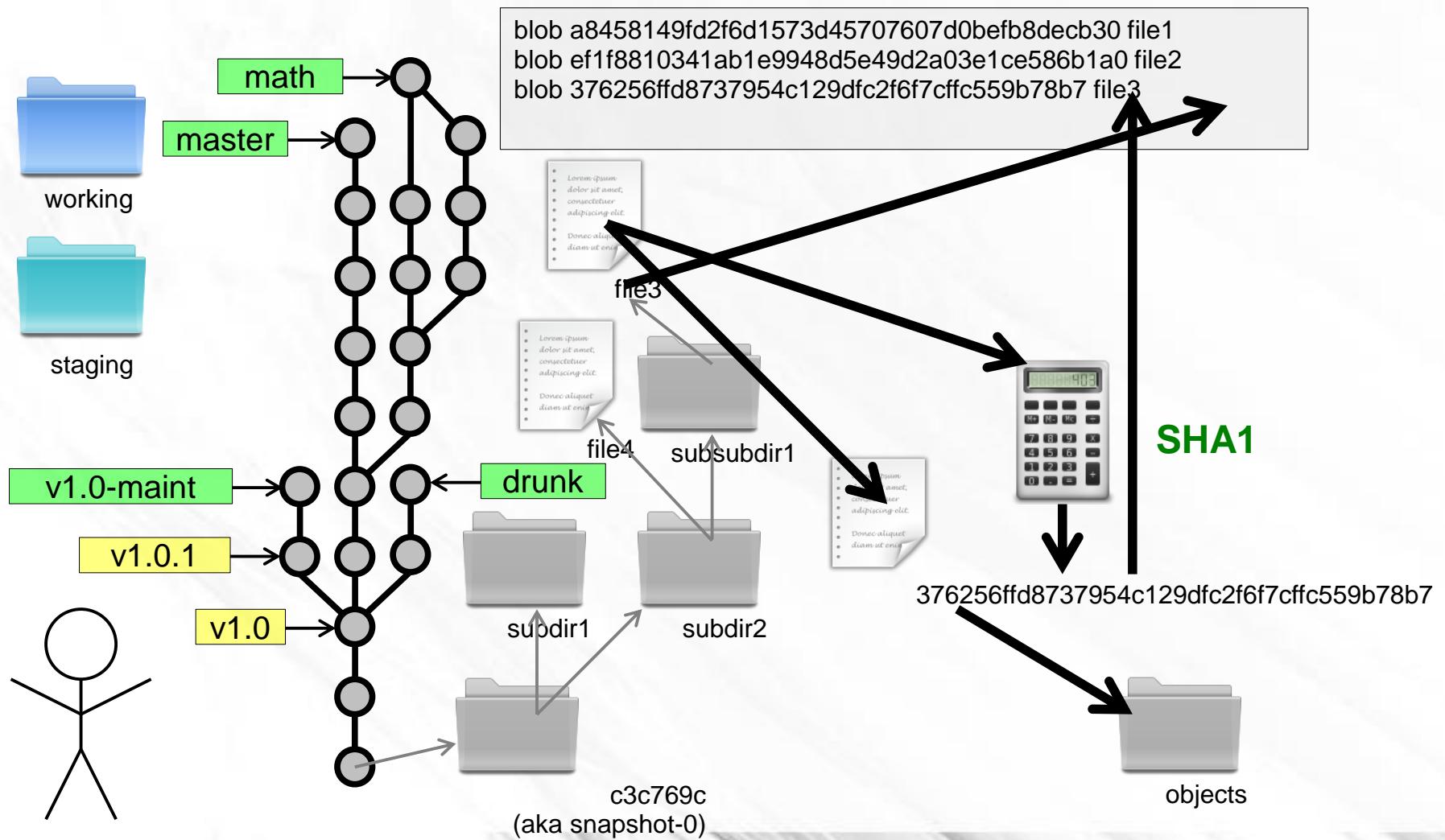
Eliminating Duplication



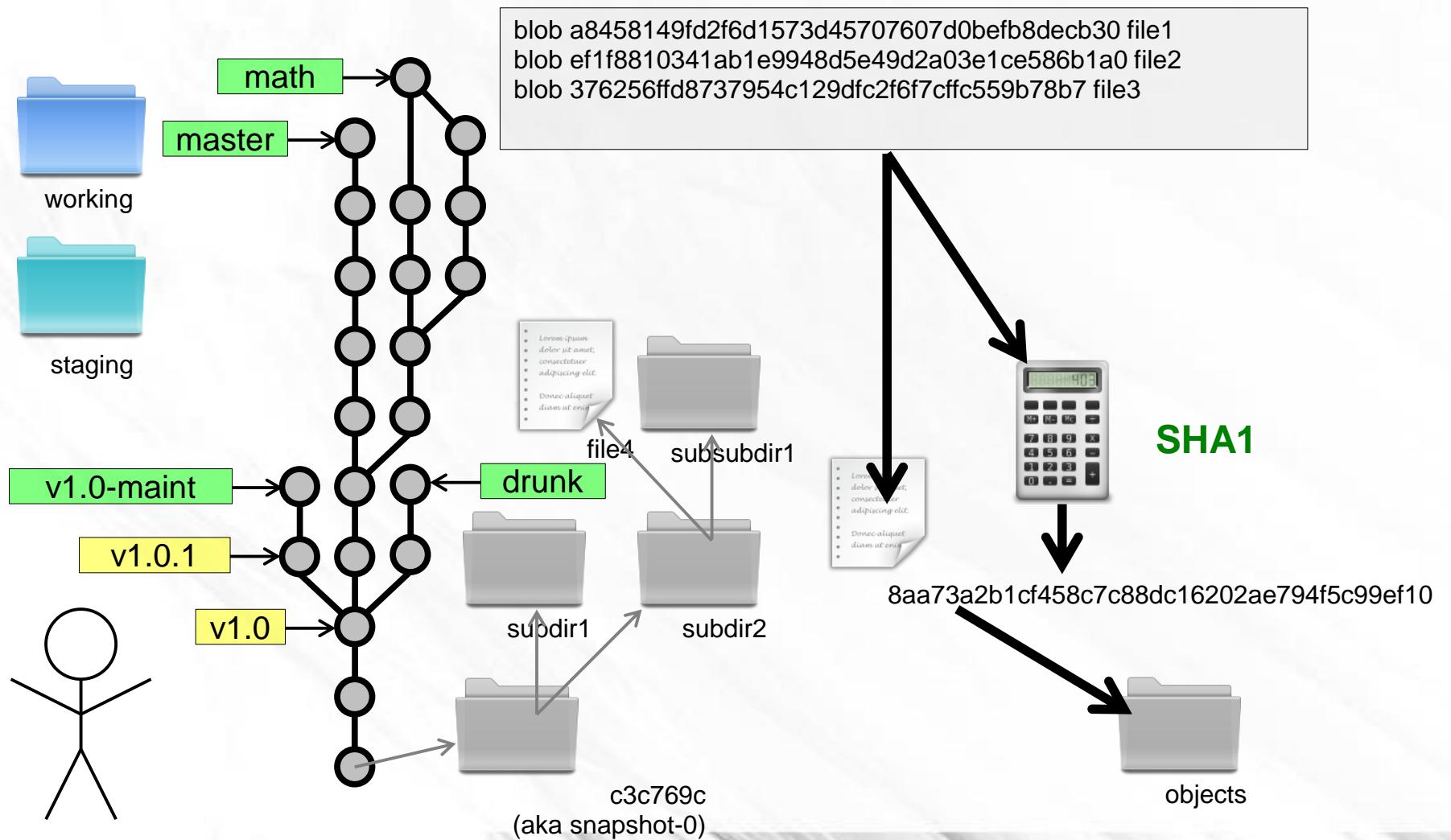
Eliminating Duplication



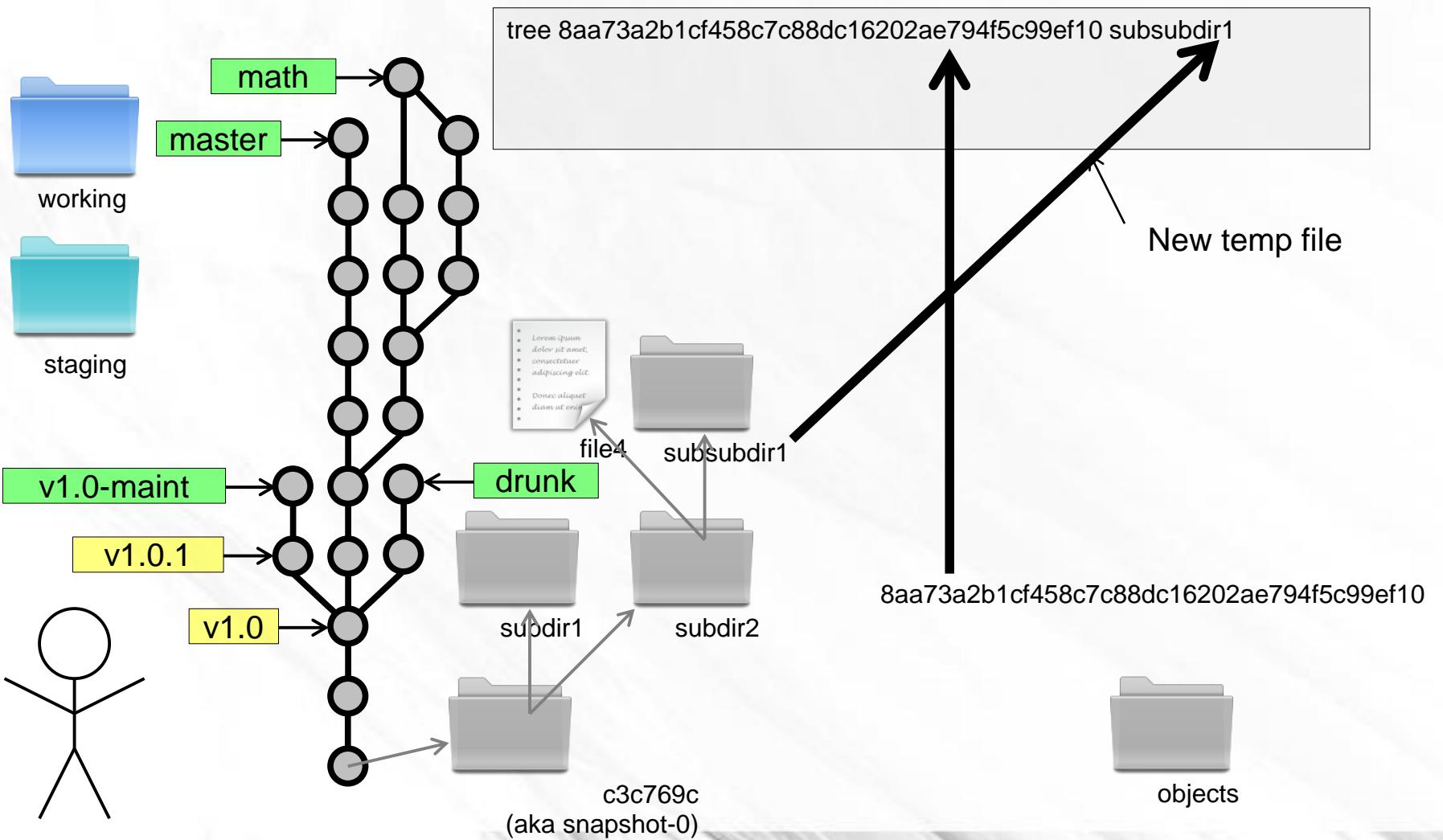
Eliminating Duplication



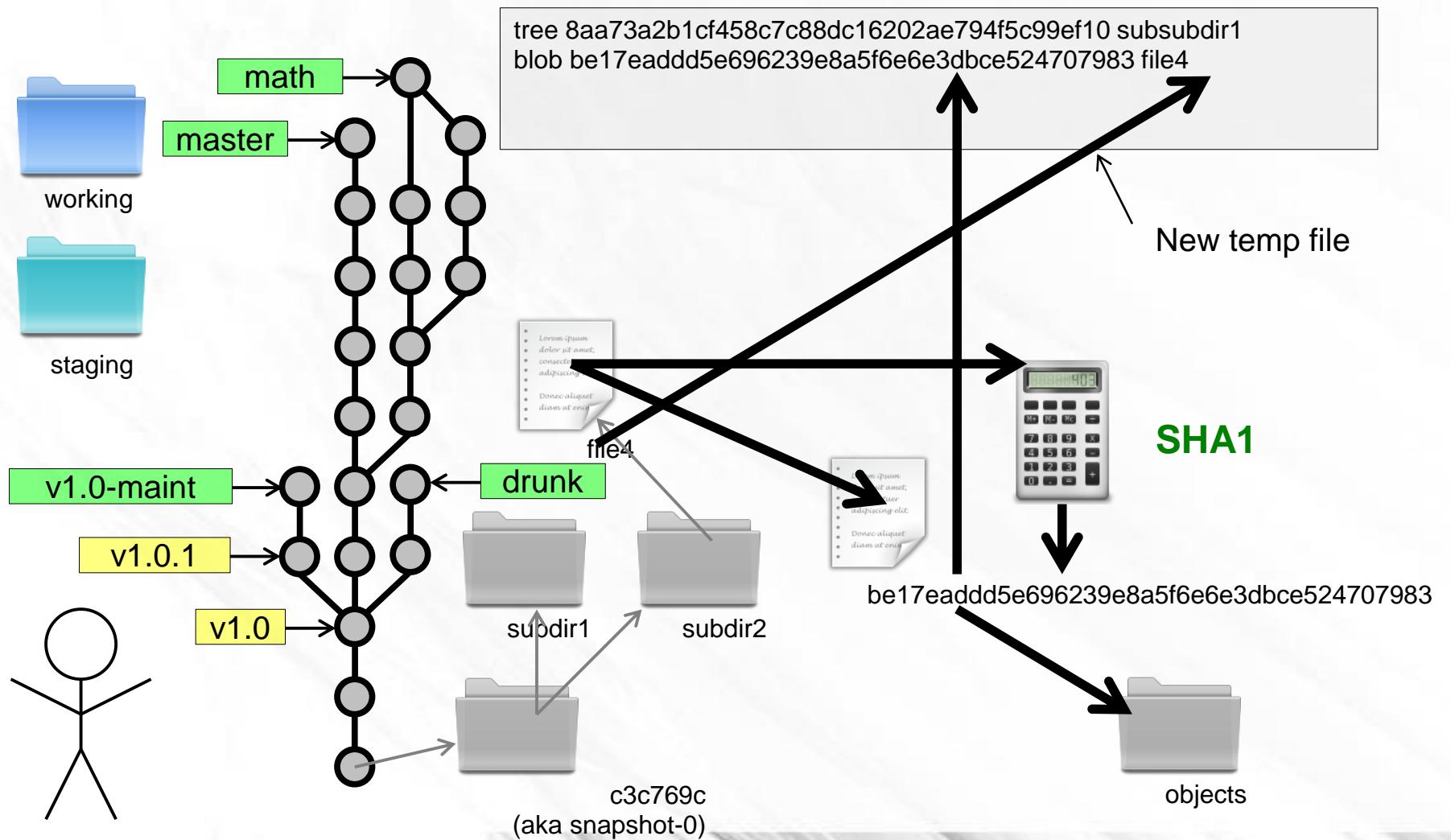
Eliminating Duplication



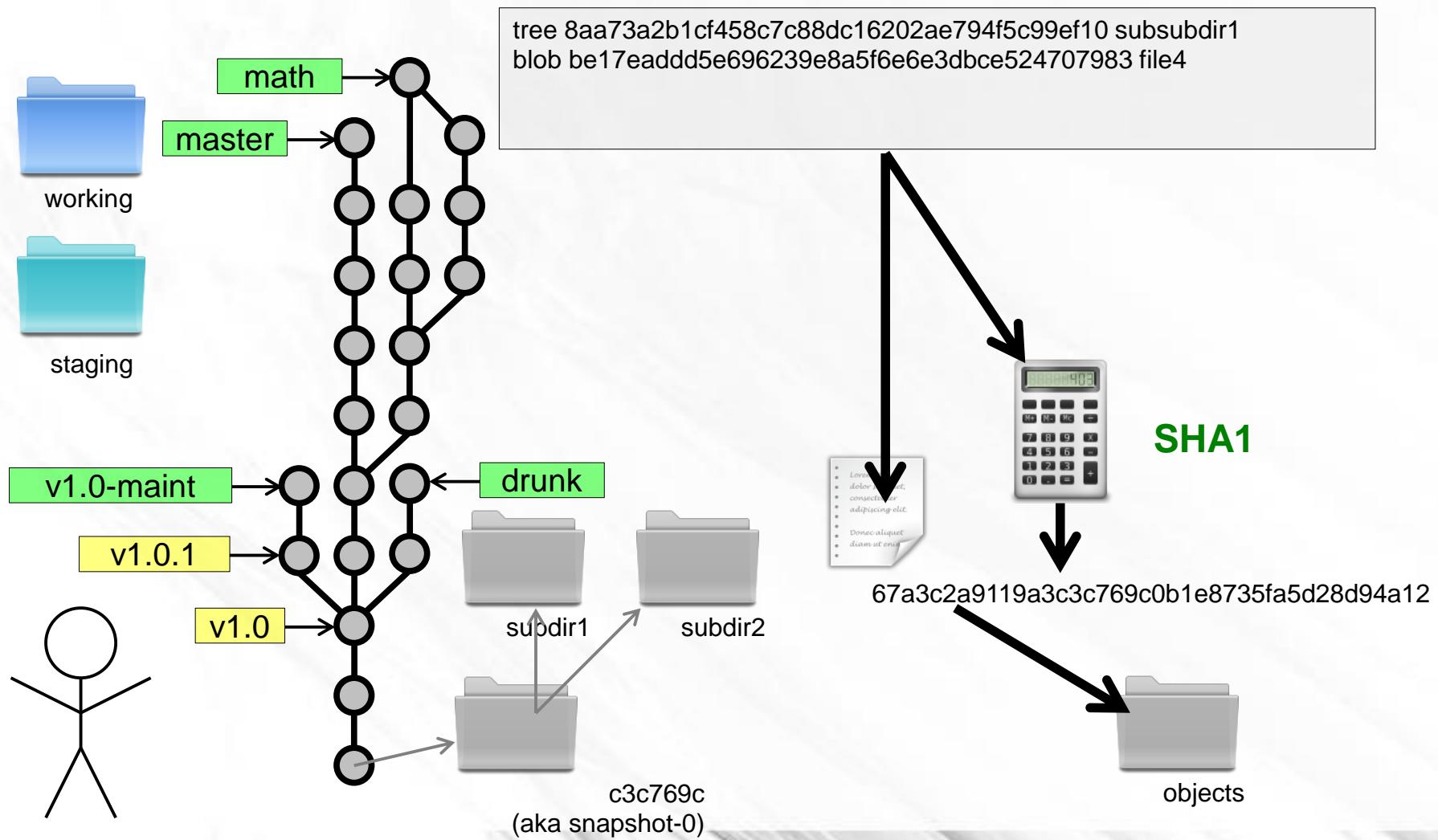
Eliminating Duplication



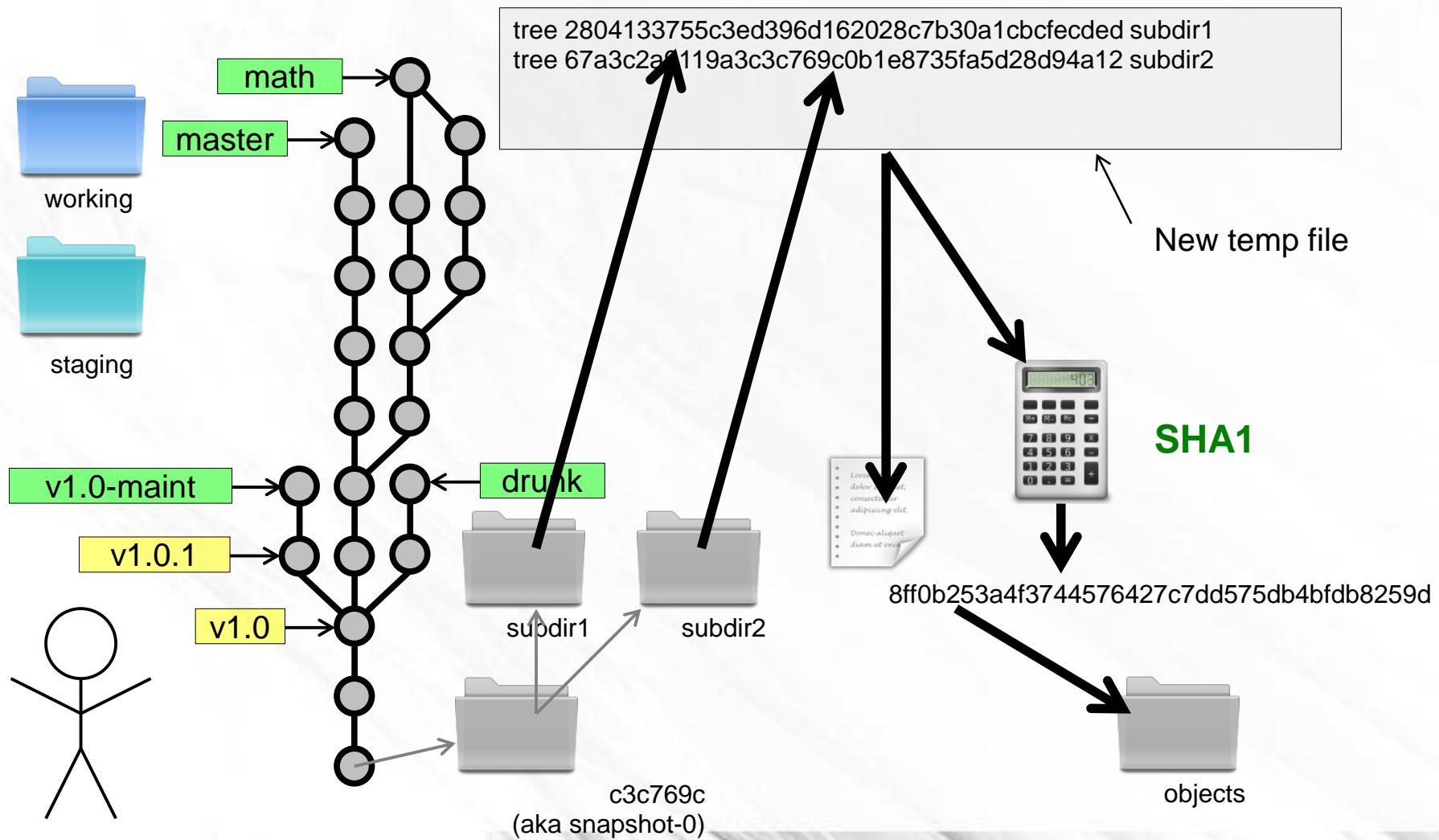
Eliminating Duplication



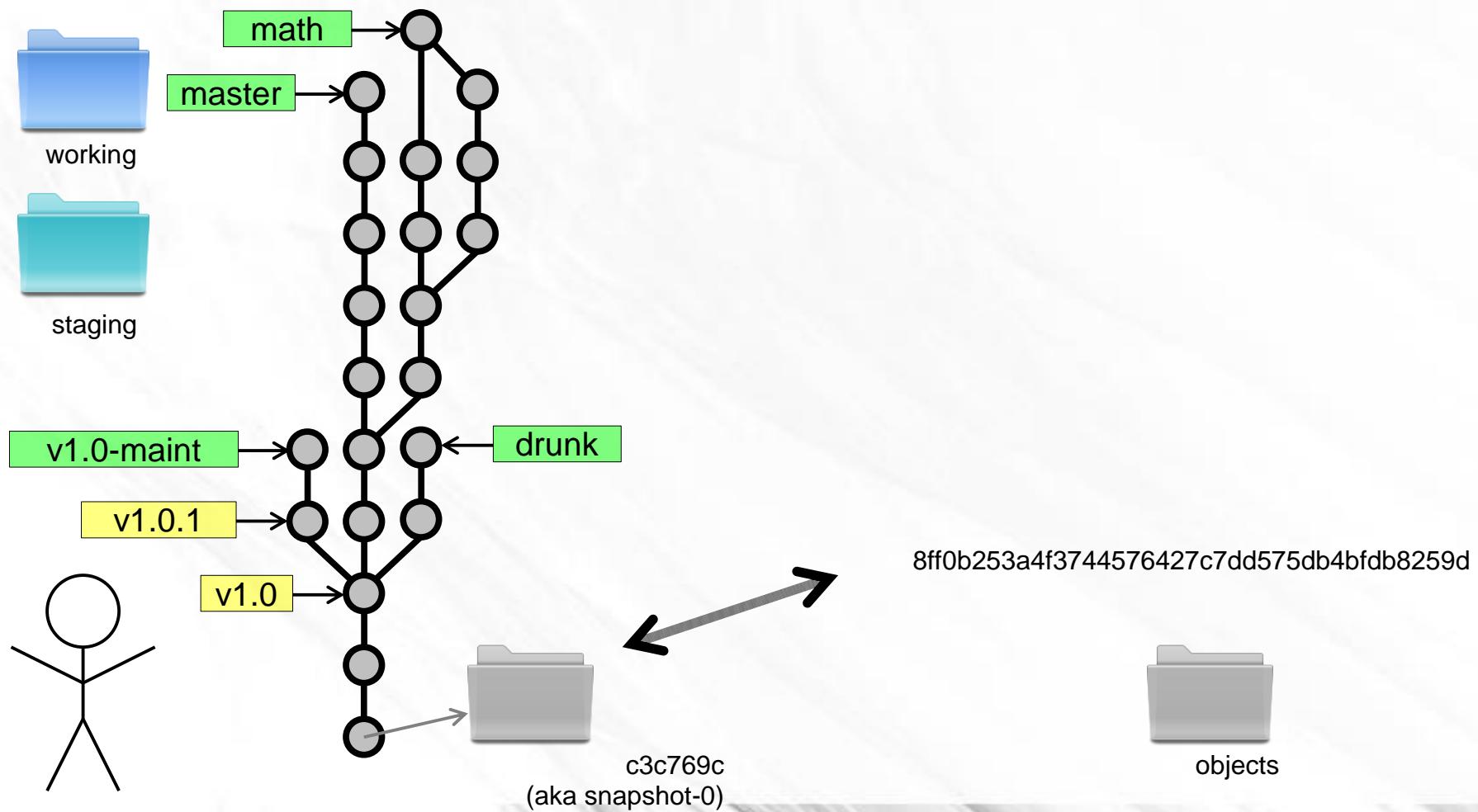
Eliminating Duplication



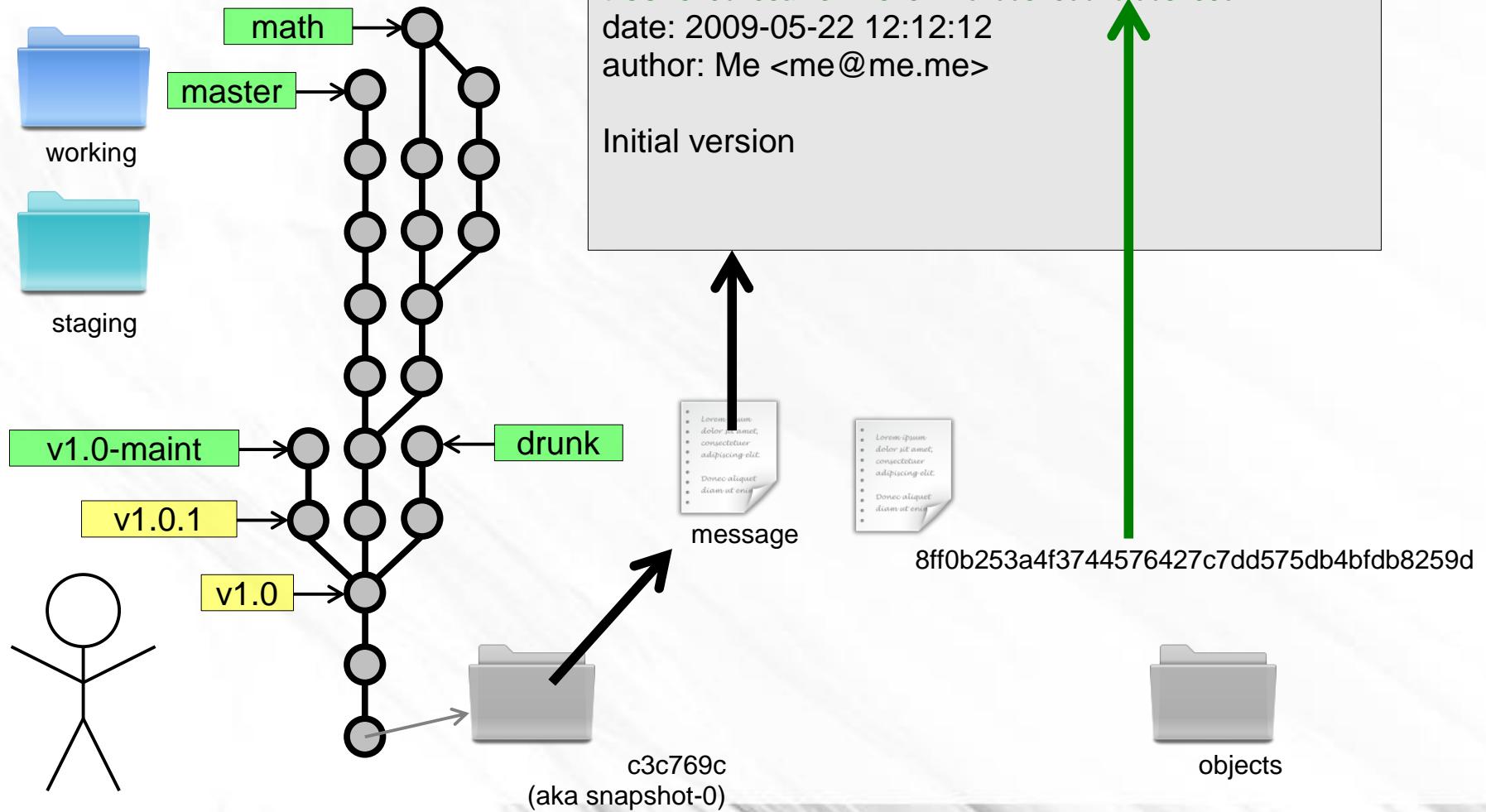
Eliminating Duplication



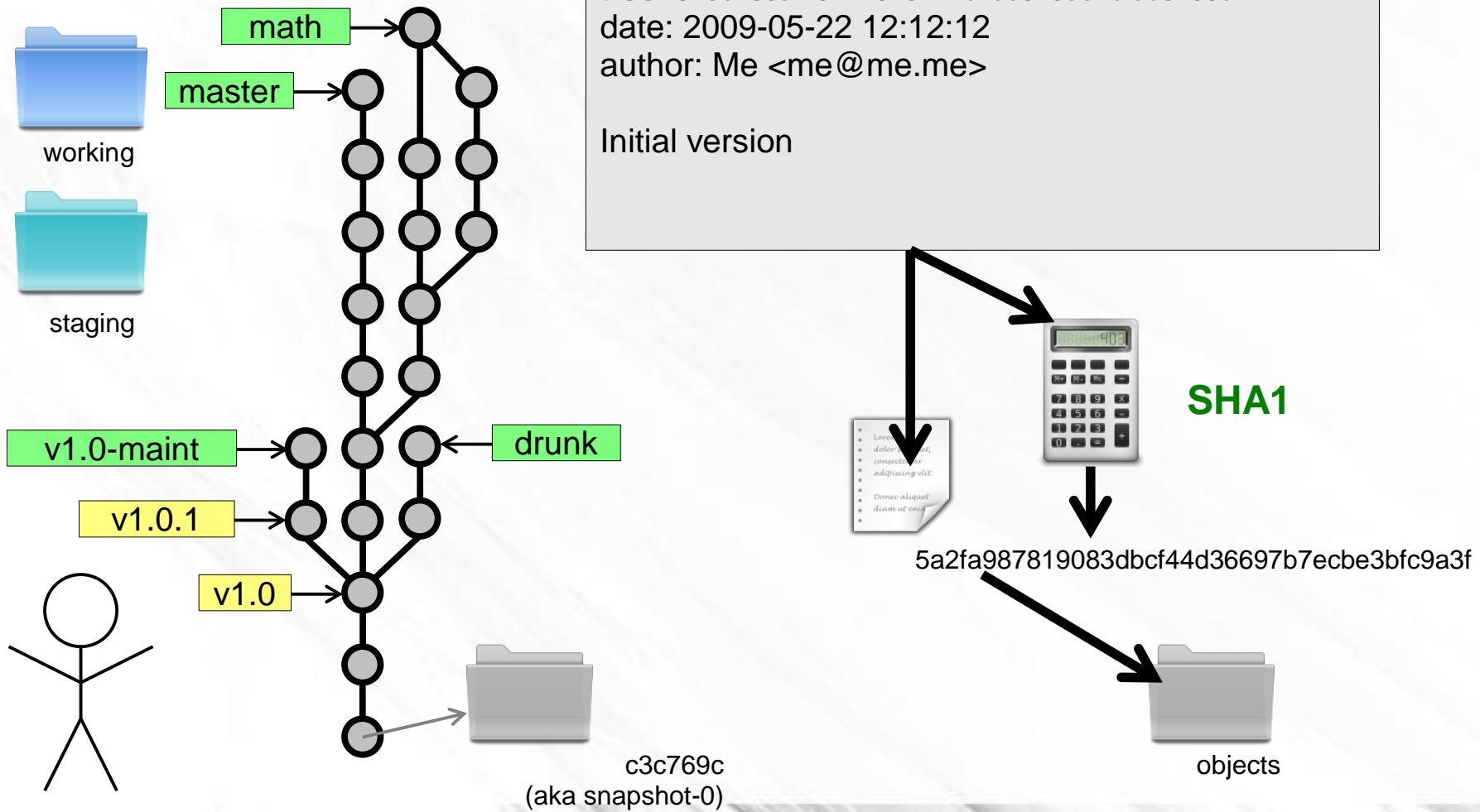
Eliminating Duplication



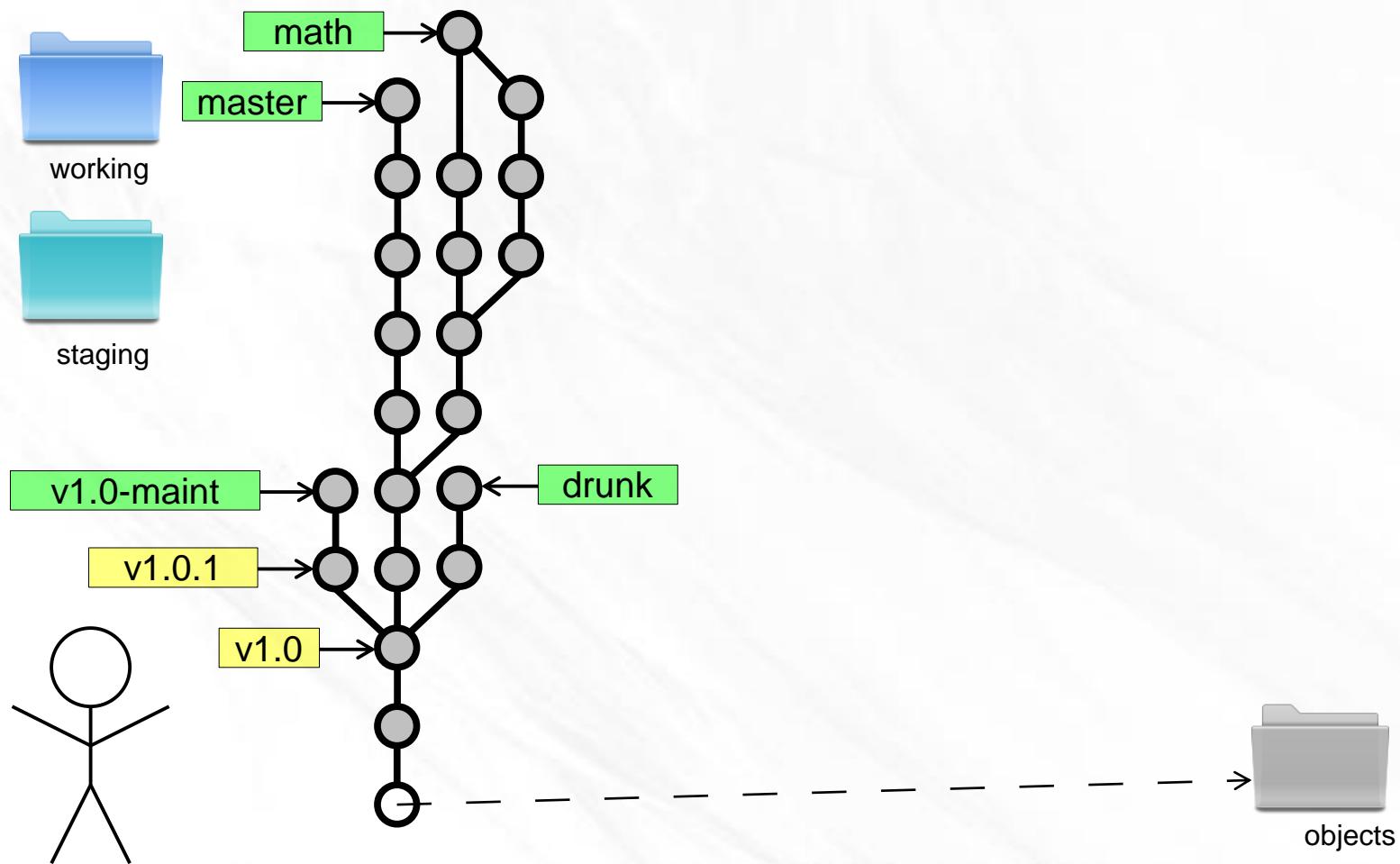
Eliminating Duplication



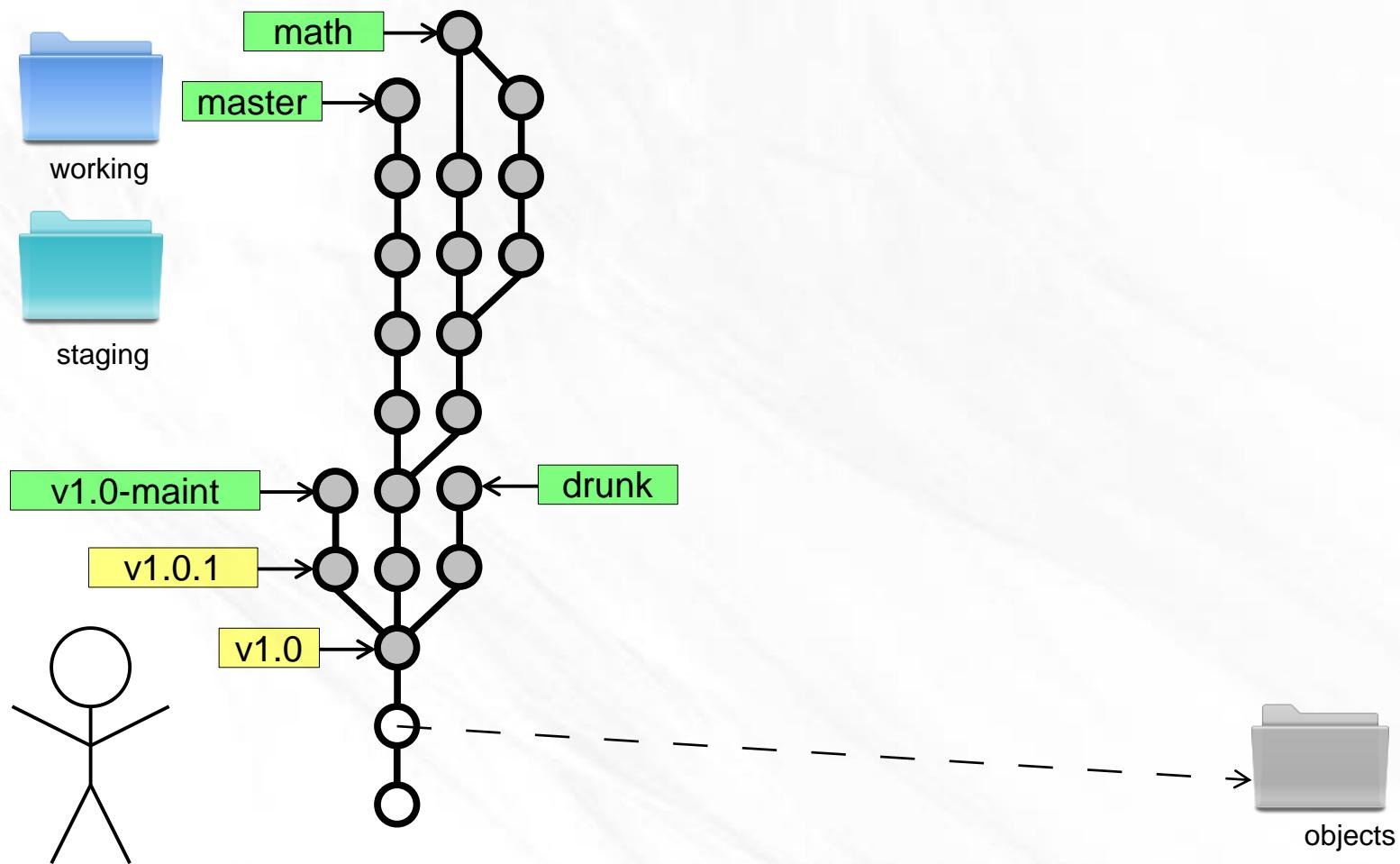
Eliminating Duplication



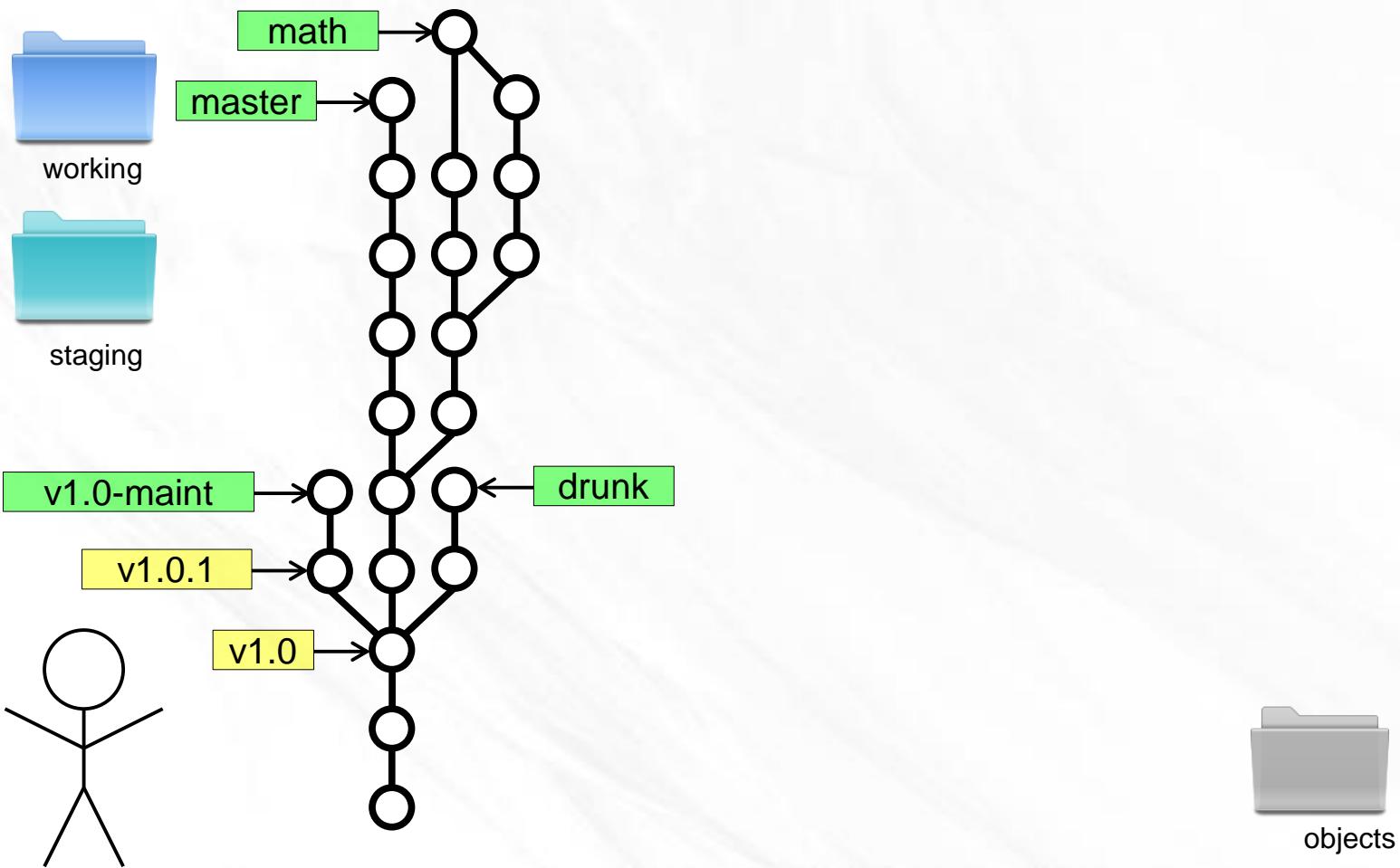
Eliminating Duplication



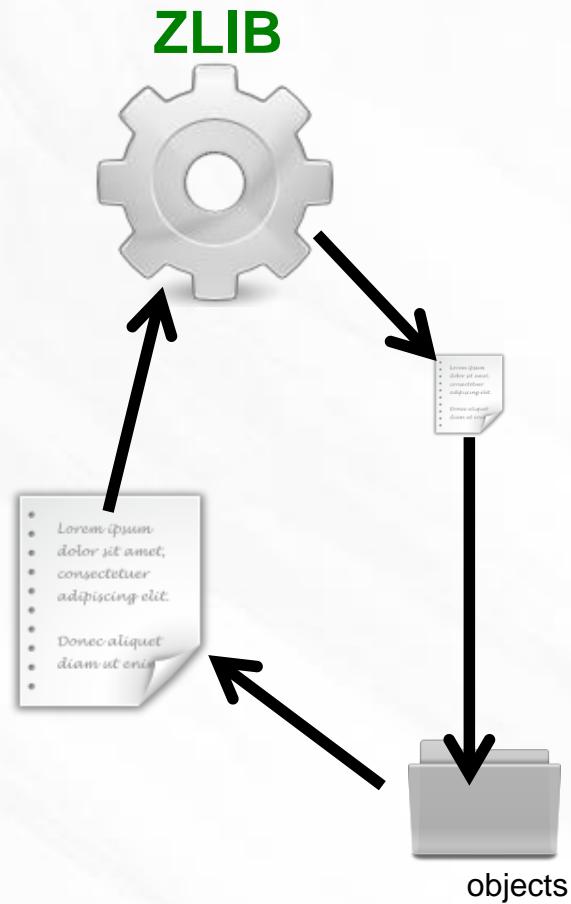
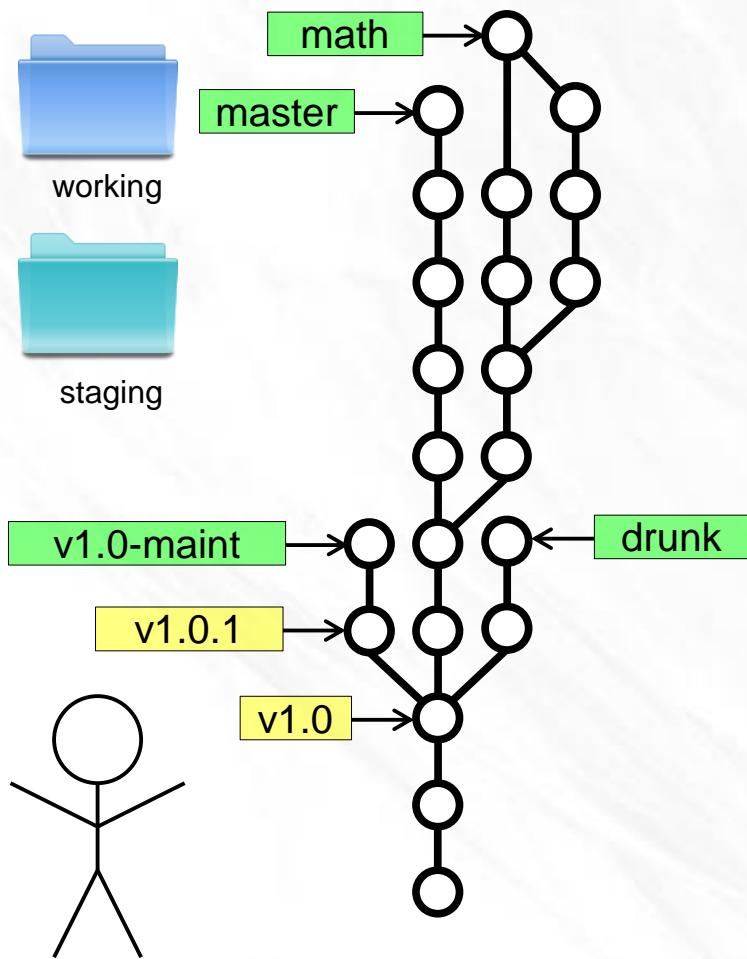
Eliminating Duplication



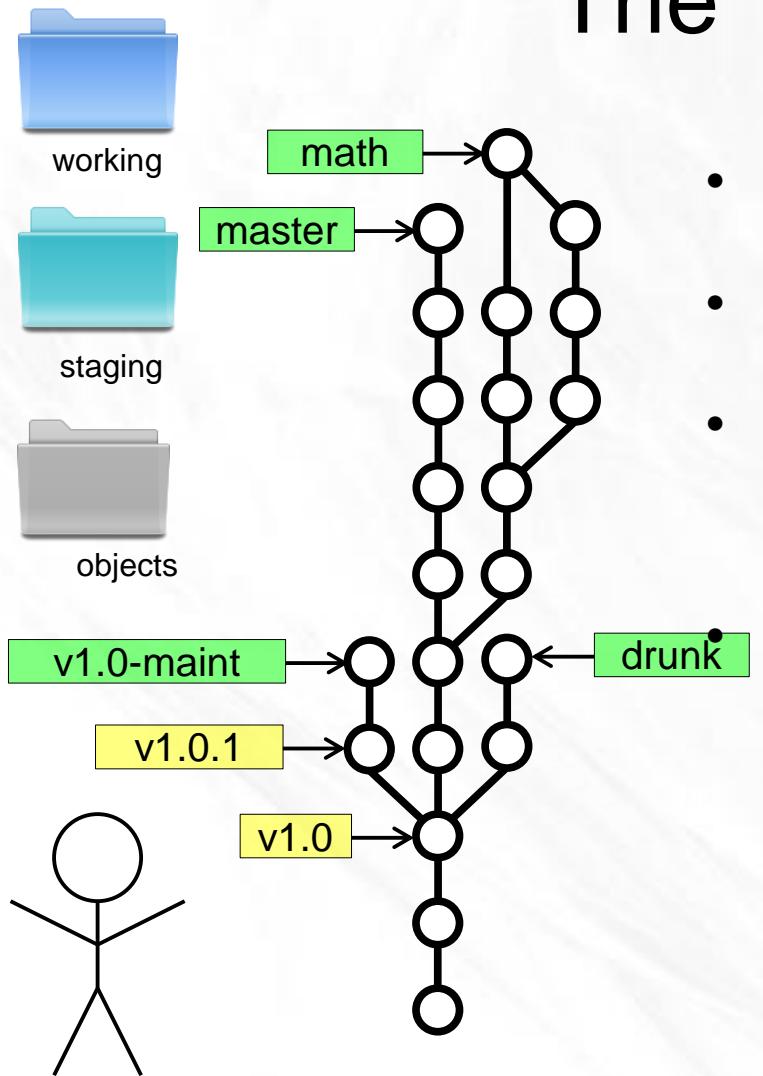
Eliminating Duplication



Compressing Blobs



The True Git

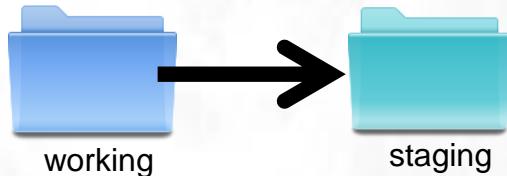


- TADAA!
 - This is pretty much Git
 - Nicer command line tools for all these operations
- Many, many other tools

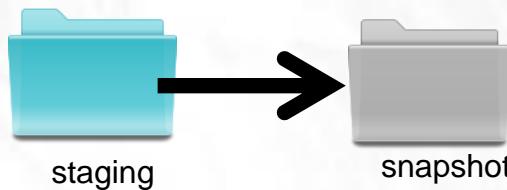
Commands: Getting Started

- First, tell Git who you are:
 - `git config --global user.name "My Name"`
 - `git config --global user.email "my@email.address"`
- Get help:
 - `git <command> -h`
 - `git help <command>`
- Start a new Git repository:
 - `git init`

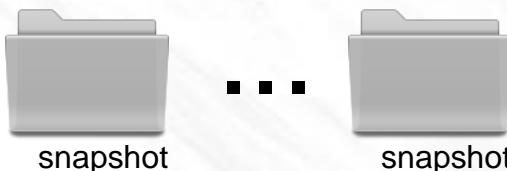
Commands: Making snapshots



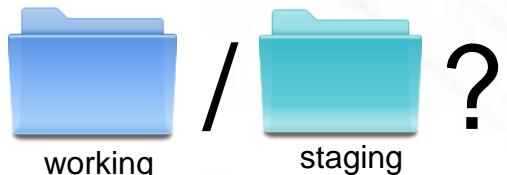
- git add



- git commit



- git log

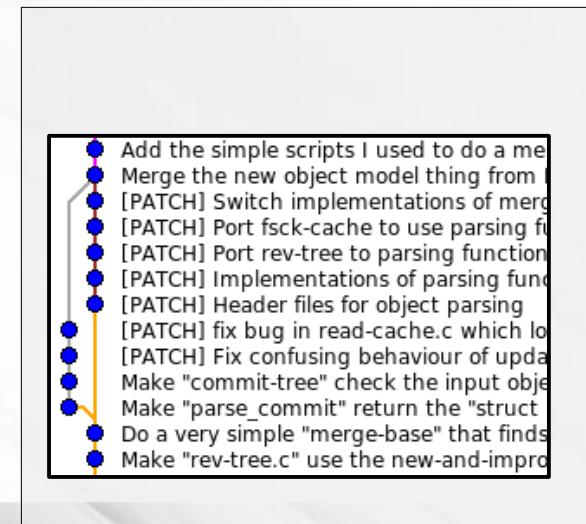


- git status

}

git commit -a

gitk



Commands: Differing



vs.



- `git diff`



vs.



- `git diff --staged`



vs.



- `git diff HEAD`



vs.



- `git diff <from> <to>`

Commands: Branches & Tags

- `git branch`
 - `git branch <branch>`
 - `git checkout <branch>`
 - `git tag -l`
 - `git tag <tag>`
- 
- } `git checkout -b ...`

Commands: Fetching & Merging

- `git remote add <name> <URL>`

- `git fetch <name>`

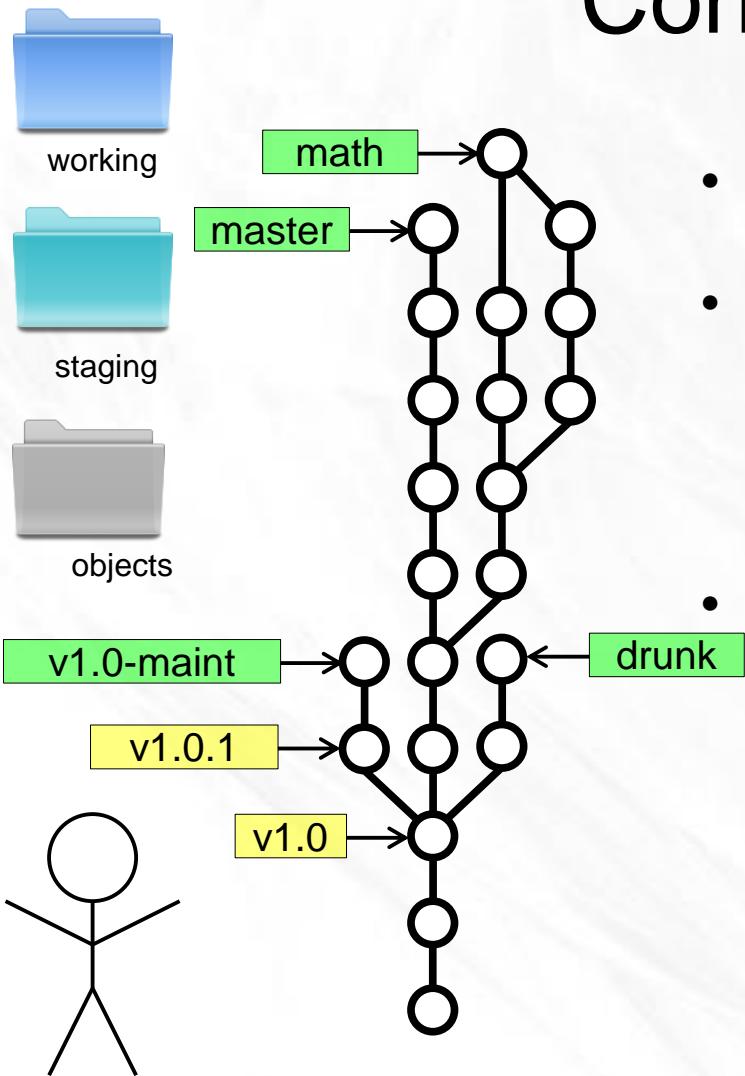
}

`git pull`

- `git merge <name>/<branch>`

- Sync back: `git push`

Conclusion



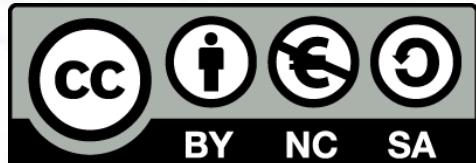
- Keep this parable in mind
- Git is simple and powerful
- One more thing:
git reflog

Where to go next? (a little outdated)

- Git homepage: <http://git-scm.com>
- Pro Git: <http://git-scm.com/book>
- Git Reference: <http://gitref.org>
- GitHub: <http://github.com>
- Gitorious: <http://gitorious.org>
- **Gitlab**

Questions?

- Thanks for your attention!
- These slides are available at:
https://github.com/jherland/git_parable
- Reach me at <johan@herland.net>



Git Clients (Windows)

- **CLI Shell: Git Bash**
- Git Extentions: CLI +Windows Explorer Shell
- Github Desktop (+powershell)
- Bitbucket SourceTree
- **IDE Integration**
 - Visual Studio (Code / VS2013+ native)
 - Eclipse Egit
 - IntelliJ, Webstorm embedded
 - Brackets plugin
 - VS Code

הדגמה \ טיפים

- Local user settings:

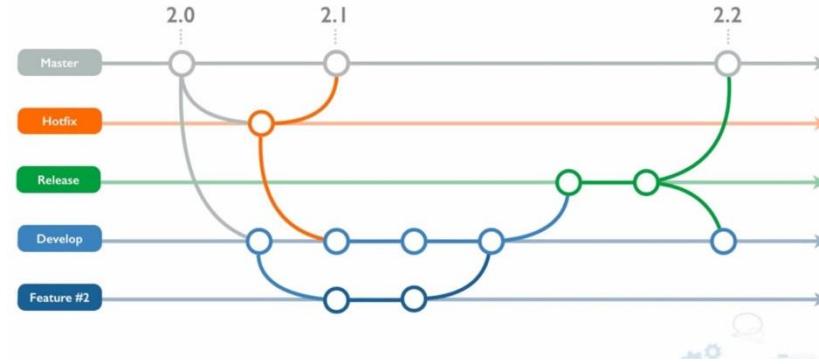
git config user.name <user>

git config user.email user@example.com

- git pull
- git add: add / stage
- git commit –a == add+commit

Git Workflows - הרחבה: שיטות עבודה

- Centralized Workflow
- Feature Branch Workflow
- Gitflow Workflow
- Forking Workflow (OSS)



Github Flow

Pull request

:Nuget דוגמא: תחילת העבודה בפרויקט
Contributing a Bug Fix or Feature

• קישורים נוספים בויקי



Git Workflow _ Atlassian Git Tutorial.pdf

- שאלות?

- הערות והארות?

- תודה רבה