

Git going with DVCS

90 minutes of mental spoon bending
via source code control

Hello
my name is

Matthew
McCullough



Matthew McCullough's git Bookmarks

[Bookmarks](#) | [Network](#) | [Tags](#) | [Subscriptions](#) | [Inbox](#)

See all git bookmarks in [Popular](#), [Recent](#), or your [Network](#).

matthew.mccullough ➤ git ✖ Type another tag

Bookmarks 118

[Display options ▾](#)

06 APR 10 [Repopular | Code Worth Mentioning - GitHub Repos](#) 34

[EDIT](#) | [DELETE](#)

opensource < programming > git

02 APR 10 [brotherbard's gitx experiment](#) 9

Experimental fork of Pieter's nice git GUI for OS X. Includes: sidebar, fetch, pull, push, add remote, merge, cherry-pick, rebase, clone, clone to.

[EDIT](#) | [DELETE](#)

macosx < opensource < freeware < sourcecodecontrol
programming < gui < git

29 MAR 10 [Git Client - SmartGit](#) 246

[EDIT](#) | [DELETE](#)

sourcecodecontrol < gui < git

[tig, the ncurses front-end to Git](#) 158

Test based GUI for Git

[EDIT](#) | [DELETE](#)

sourcecodecontrol < opensource < gui < git

[NBGit - Git Support for NetBeans](#) 64

[EDIT](#) | [DELETE](#)

netbeans < sourcecodecontrol < plugin < gui < opensource < git

28 MAR 10 [A tour of git: the basics](#) 651

[EDIT](#) | [DELETE](#)

sourcecodecontrol < howto < tutorial < programming < git

- [Save a new bookmark](#)
- [Create tag description](#)
- [Bulk edit](#)
- [Browse these bookmarks BETA](#)

Tags

People

Options ▾

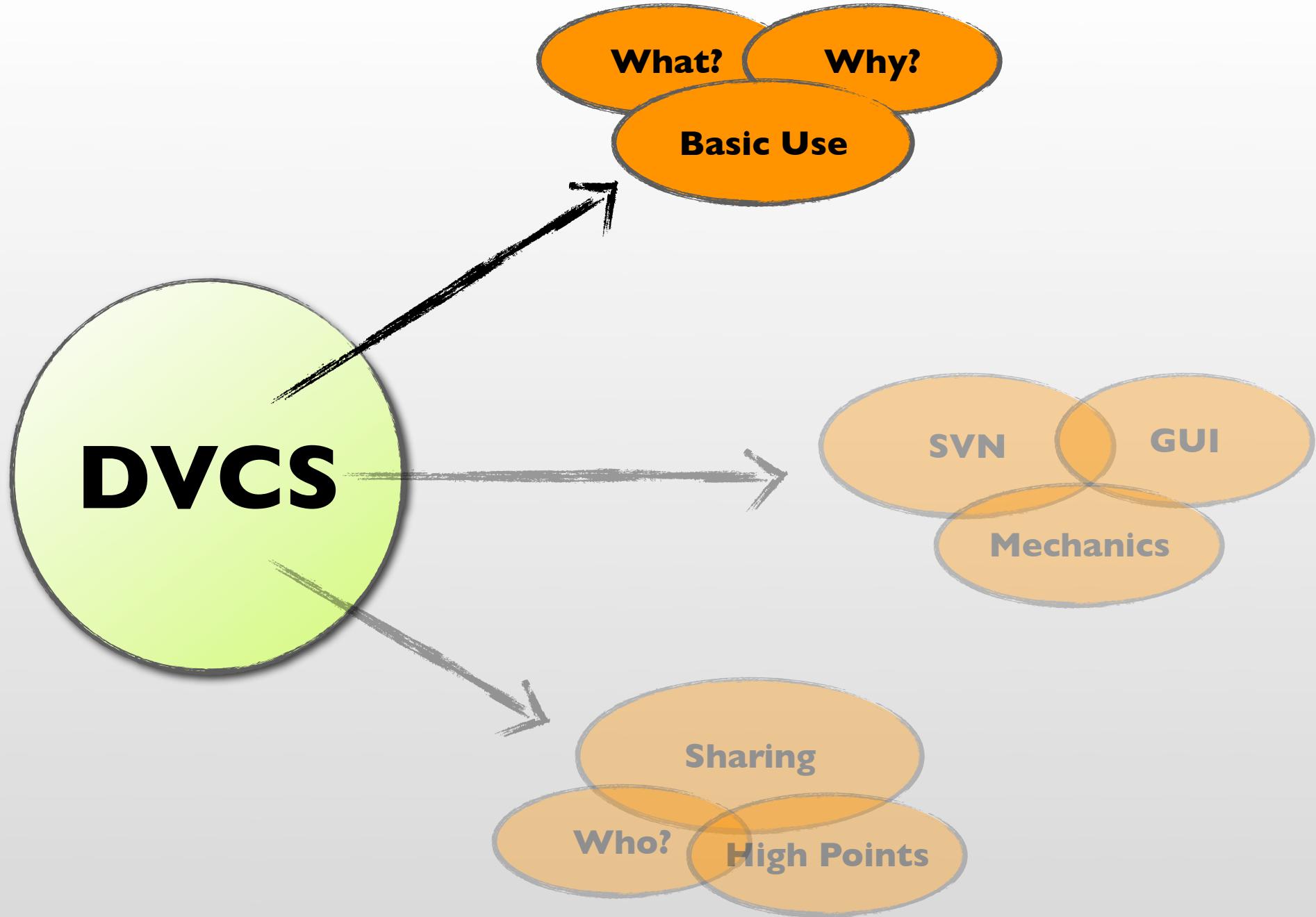
Related Tags 77

+agile	2
+animation	1
+apache	1
+bash	1
+blog	5
+bugtracking	3
+cheatsheet	10
+clearcase	1
+cocoa	1
+collaboration	1
+comics	1
+continuousintegration	2
+converter	2
+development	1
+documentation	6
+download	1
+dvcs	4
+git	2
+eclipse	2
+education	3
+free	1
+freeware	1

<http://delicious.com/matthew.mccullough/git>



<http://github.com/matthewmccullough/git-workshop>



ait

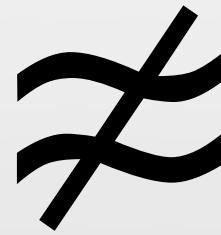
"cool kids" version control system

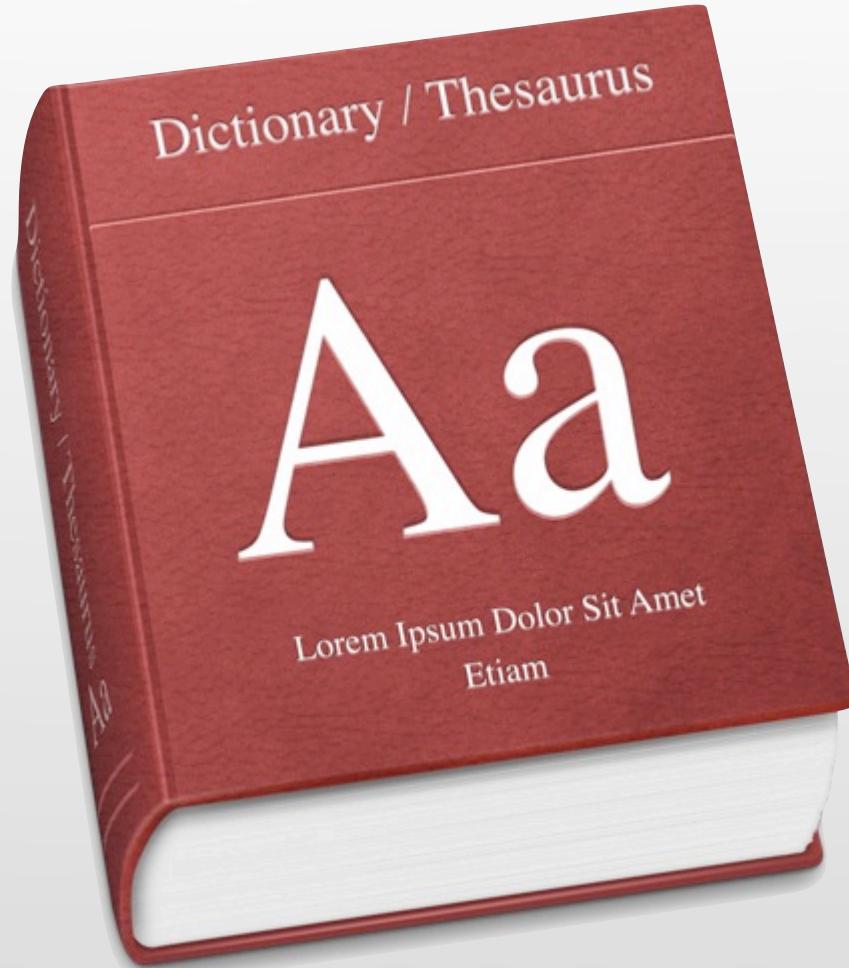
Open Source

bash scripts



C code





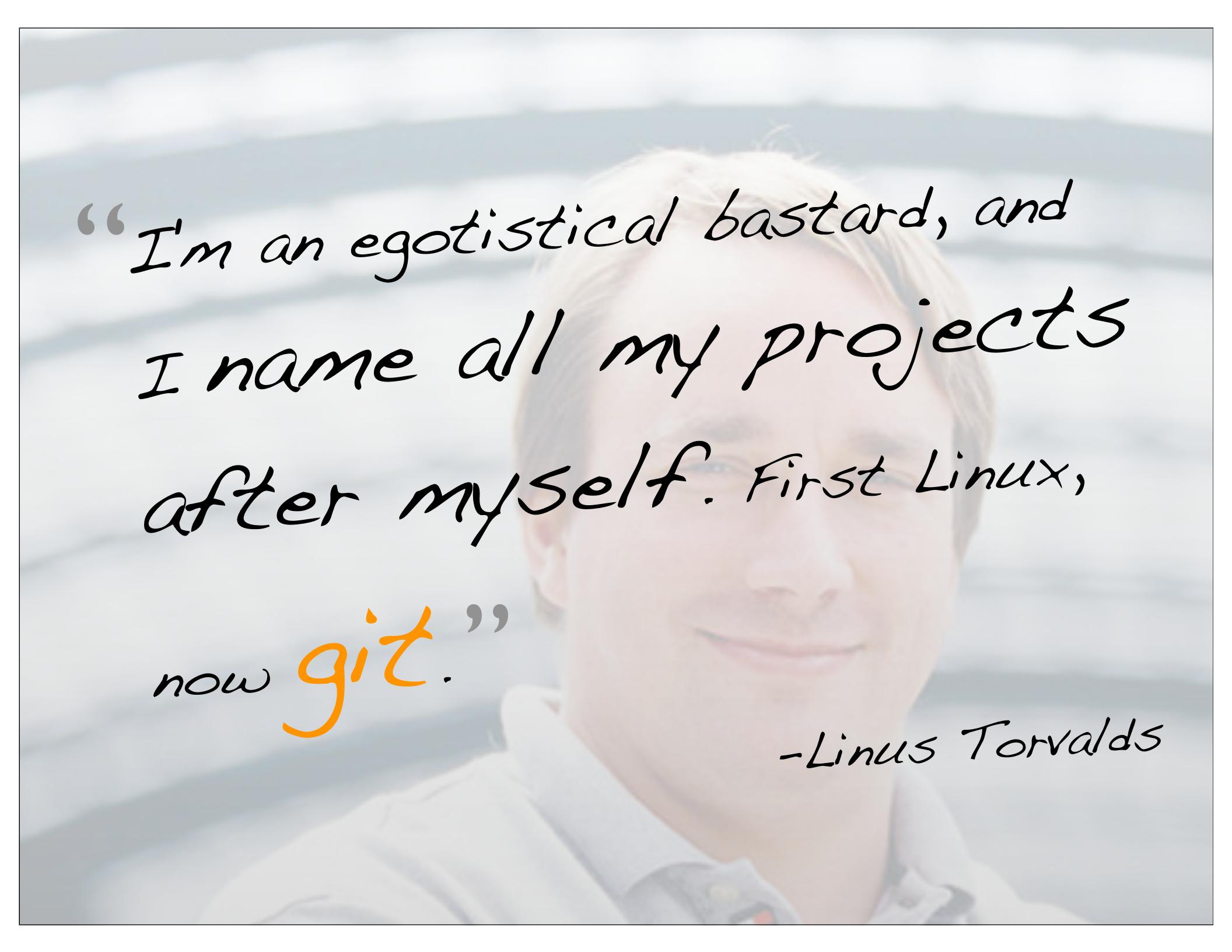
“**Git**

-noun

British Slang. an **unpleasant**,
or contemptible person”

-Oxford English Dictionary



A portrait of Linus Torvalds, a man with light brown hair and a slight smile, wearing a white shirt.

“I’m an egotistical bastard, and
I name all my projects
after myself. First Linux,
now git.”

-Linus Torvalds



① Download binaries

Git Distro Sources

Win Cygwin

Win msysGit

Linux Source

Linux Package manager

Mac MacPorts

Mac git-osx-installer

Any JGit

② Put it in your PATH

③

Identify yourself

Global Git Settings

```
git config --global user.name "Hal Smith"
```

```
git config --global user.email "hal@bnl.com"
```

④

USE IT!

CREATING A REPO

`mkdir myproj.git`

`cd myproj.git`

`git init`

Certified
Git User

Command Terminology

Plumbing is the low level utilities

Porcelain is the set of *high-level* commands

Porcelain is comprised of plumbing

High-level commands (porcelain)

We separate the porcelain commands into the main commands and some ancillary user utilities.

Main porcelain commands

[git-add\(1\)](#)

Add file contents to the index.

[git-am\(1\)](#)

Apply a series of patches from a mailbox.

[git-archive\(1\)](#)

Create an archive of files from a repository.

[git-bisect\(1\)](#)

Fix bugs.

[git-bran...](#)

List branches.

[git-bund...](#)

Merge.

[git-check...](#)

Check.

[git-cherr...](#)

Applies patches in the working tree and.

[git-citool...](#)

Gr...

[git-clean\(1\)](#)

Re...

[git-clone\(1\)](#)

Cl...

[git-comm...](#)

Re...

[git-descri...](#)

Show

[git-dif...](#)

Show

[git-fetch\(1\)](#)

Dow...

[git-format...](#)

Pre...

[git-gc\(1\)](#)

Clea...

[git-grep\(1\)](#)

Print...

[git-gui\(1\)](#)

A po...

[git-init\(1\)](#)

Cr...

[git-log\(1\)](#)

Show

[git-merge\(1\)](#)

Join t...

[git-mv\(1\)](#)

Move

[git-pull\(1\)](#)

Fetch

[git-push\(1\)](#)

Update

Interrogation commands

[git-cat-file\(1\)](#)

Provide content or type/size information.

[git-diff-files\(1\)](#)

Compares files in the working tree and.

[git-diff-index\(1\)](#)

Compares content and mode of blobs.

[git-diff-tree\(1\)](#)

Compares the content and mode of b...

[git-for-each-ref\(1\)](#)

Output information on each ref.

[git-ls-files\(1\)](#)

Show information about files in the i...

[git-ls-remote\(1\)](#)

List references in a remote repository.

[git-ls-tree\(1\)](#)

List the contents of a tree object.

[git-merge-base\(1\)](#)

Find as good common ancestors as

[git-name-rev\(1\)](#)

Find symbolic names for given revs

[git-pack-redundant\(1\)](#)

Find redundant pack files.

[git-rev-list\(1\)](#)

Lists commit objects in reverse chro...

[git-show-index\(1\)](#)

Show packed archive index.

[git-show-ref\(1\)](#)

List references in a local repository

[git-tar-tree\(1\)](#)

(deprecated) Create a tar archive o...

[git-unpack-file\(1\)](#)

Creates a temporary file with a blo...

[git-var\(1\)](#)

Show a git logical variable.

[git-verify-pack\(1\)](#)

Validate packed git archive files.

In general, the interrogate commands do not touch the files in the working tree.

Interacting with Others

interact with foreign SCM and with other people via patch over e-mail.

position

commit

out c

Low-level commands (plumbing)

Although git includes its own porcelain layer, its low-level commands are sufficient to support development of alternative porcelains. Developers of such porcelains might start by reading about [git-update-index\(1\)](#) and [git-read-tree\(1\)](#).

The interface (input, output, set of options and the semantics) to these low-level commands are meant to be a lot like these commands are primarily for scripted use. The details are subject to change in order to improve the end user

commands into commands that manipulate objects (in the

that interrogate and compare objects, and commands that move

king tree.

blob from a file.

archive.

xt.

ck files.

the index.

y.

Synching repositories

[git-daemon\(1\)](#)

A really simple server for git repositories.

[git-fetch-pack\(1\)](#)

Receive missing objects from another repository.

[git-send-pack\(1\)](#)

Push objects over git protocol to another repository.

[git-update-server-info\(1\)](#)

Update auxiliary info file to help dumb servers.

The following are helper programs used by the above; end users typically do not use them directly.

[git-http-fetch\(1\)](#)

Download from a remote git repository via HTTP.

[git-http-push\(1\)](#)

Push objects over HTTP/DAV to another repository.

[git-parse-remote\(1\)](#)

Routines to help parsing remote repository access parameters.

[git-receive-pack\(1\)](#)

Receive what is pushed into the repository.

[git-shell\(1\)](#)

Restricted login shell for GIT-only SSH access.

[git-upload-archive\(1\)](#)

Send archive back to git-archive.

[git-upload-pack\(1\)](#)

Send objects packed back to git-fetch-pack.

[git-write-tree\(1\)](#)

Create a tree object from the current index.

| 45+ commands

I 8 used on
a daily basis

init

checkout

bisect

log

revert

clone

rebase

pack

diff

merge

pull

grep

commit

push

tag

add

status

branch

Follow THE
DOTS

Centralized

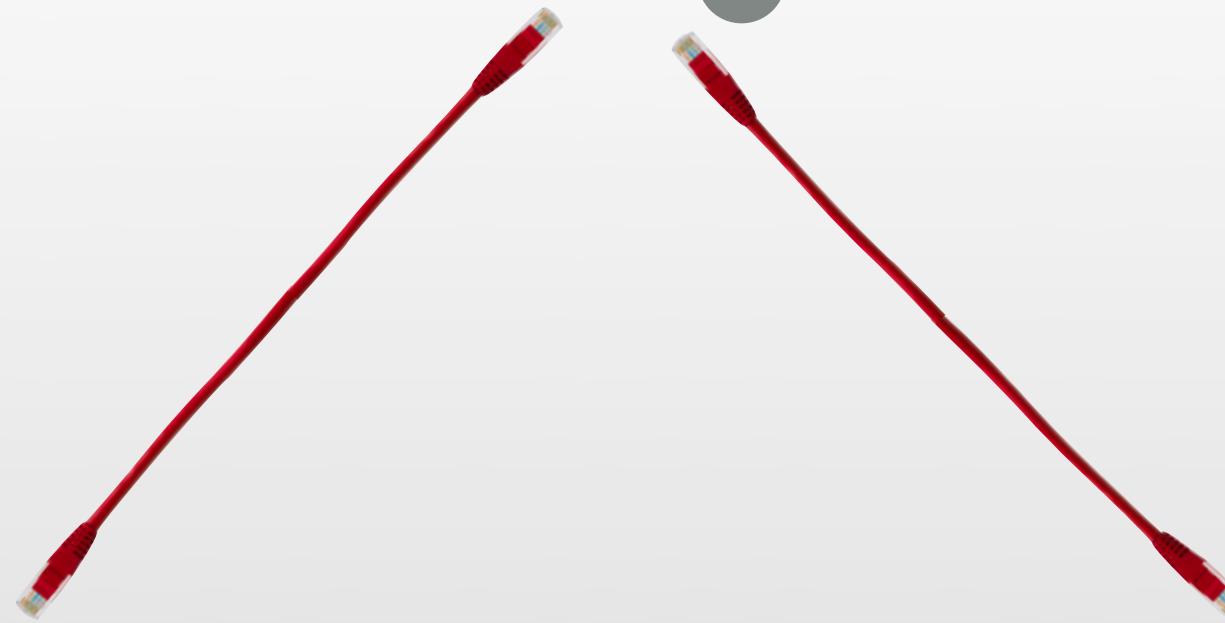
subversion server

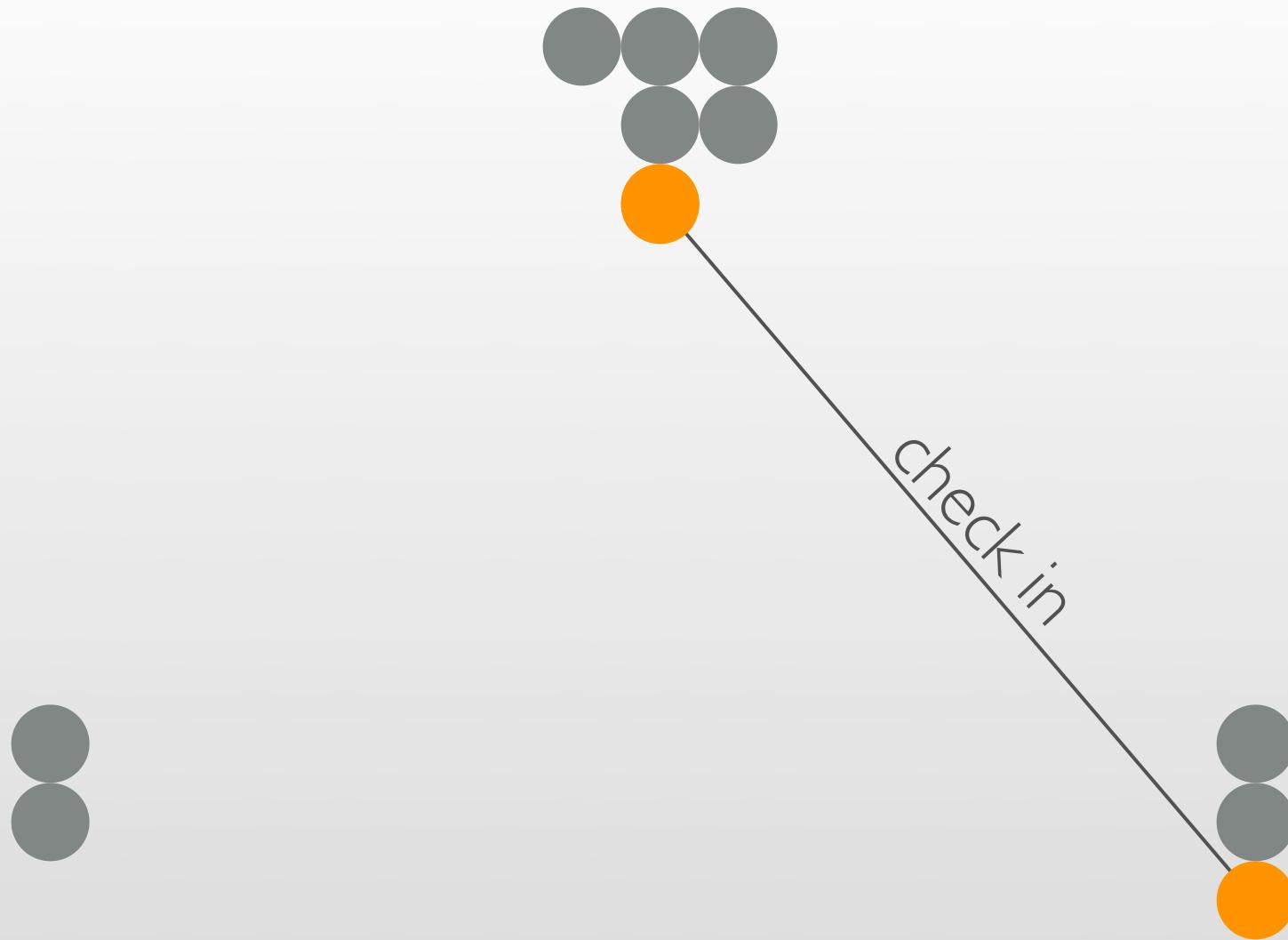


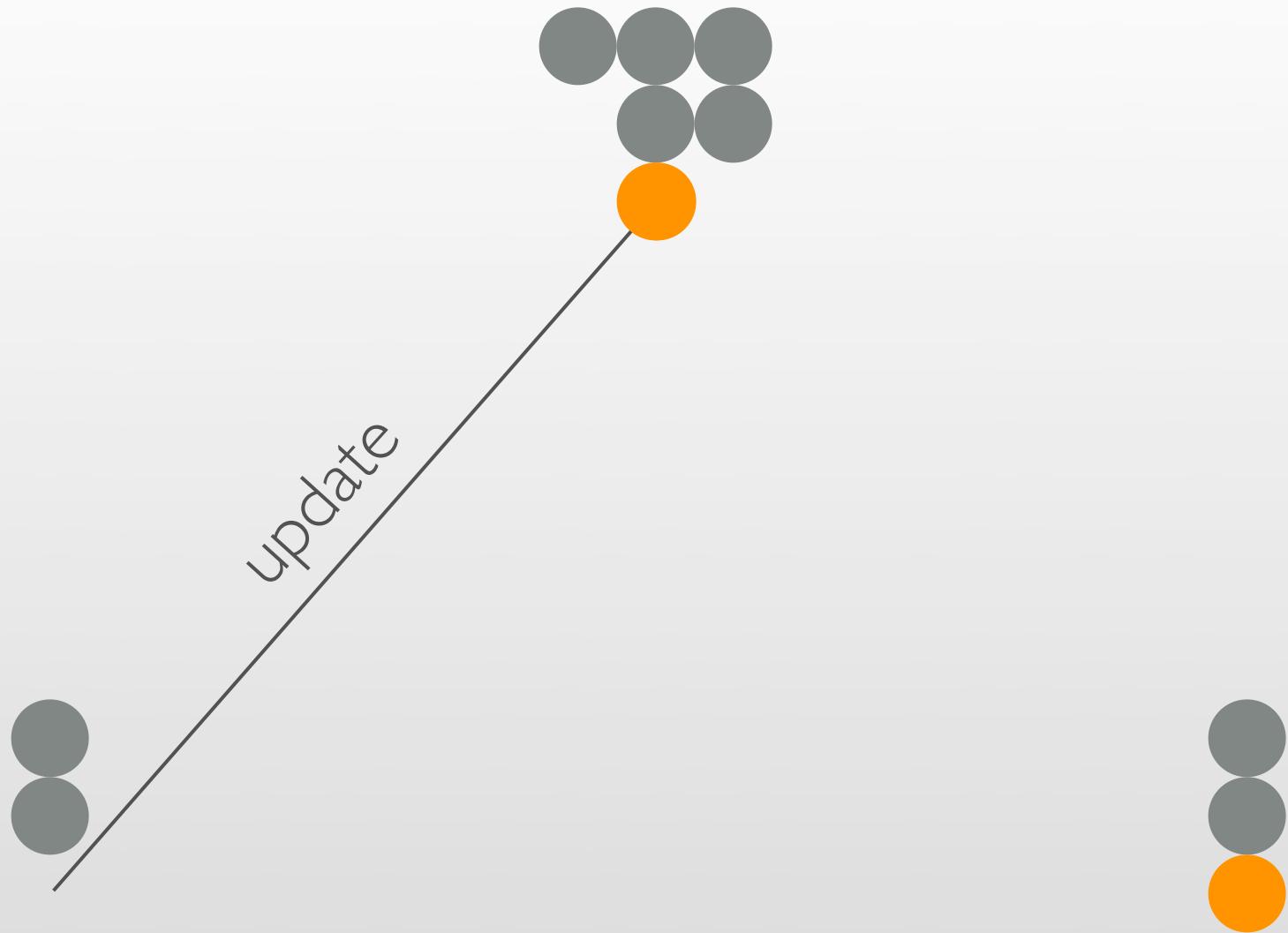
developer A

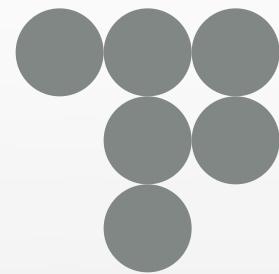


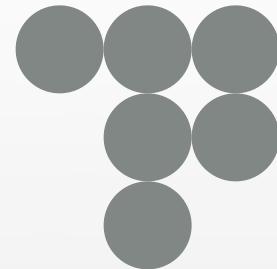
developer B











Centralized



Distributed
connectivity



developer A



developer A



developer B

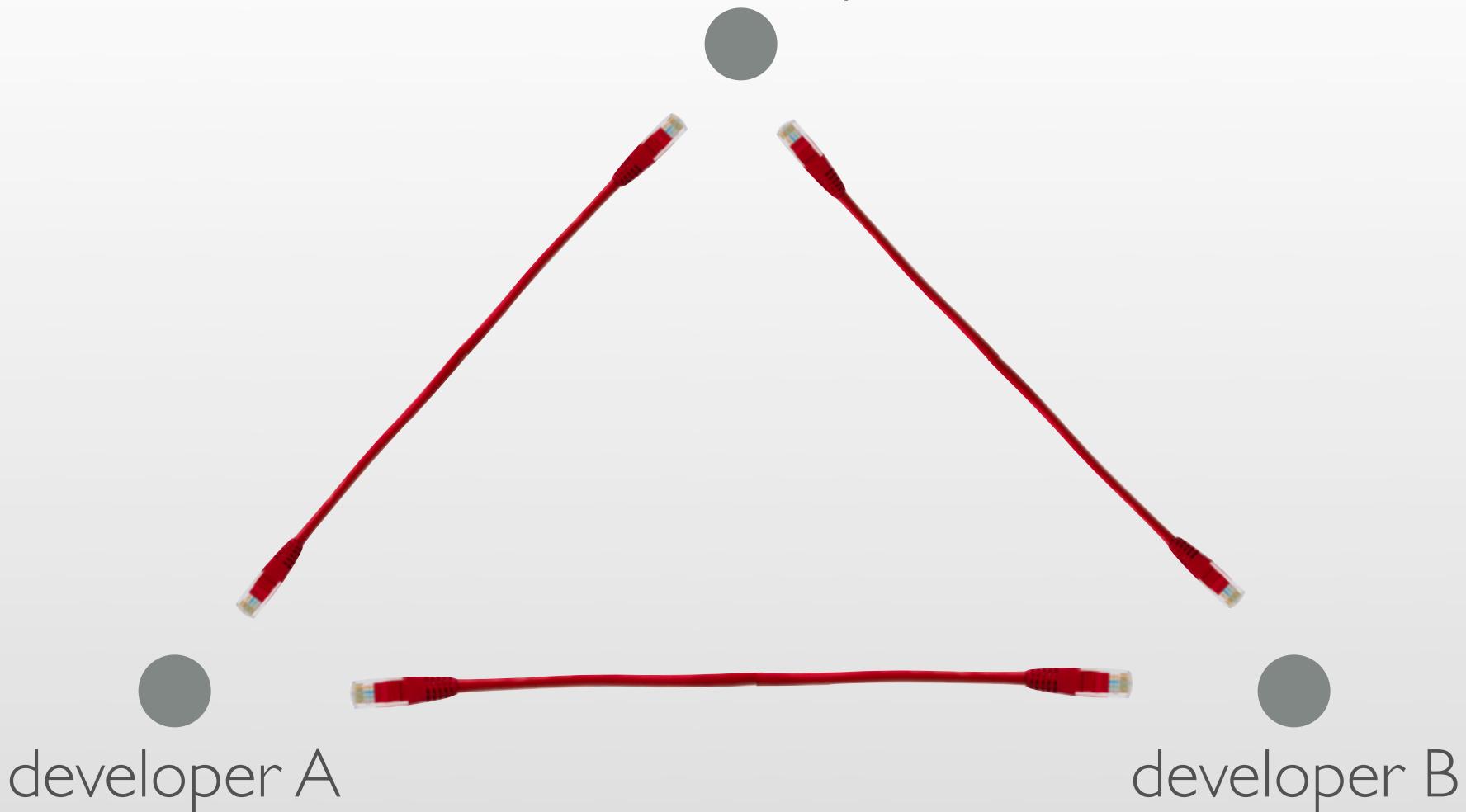


developer A



developer B

blessed repo



Distributed **commit flow**



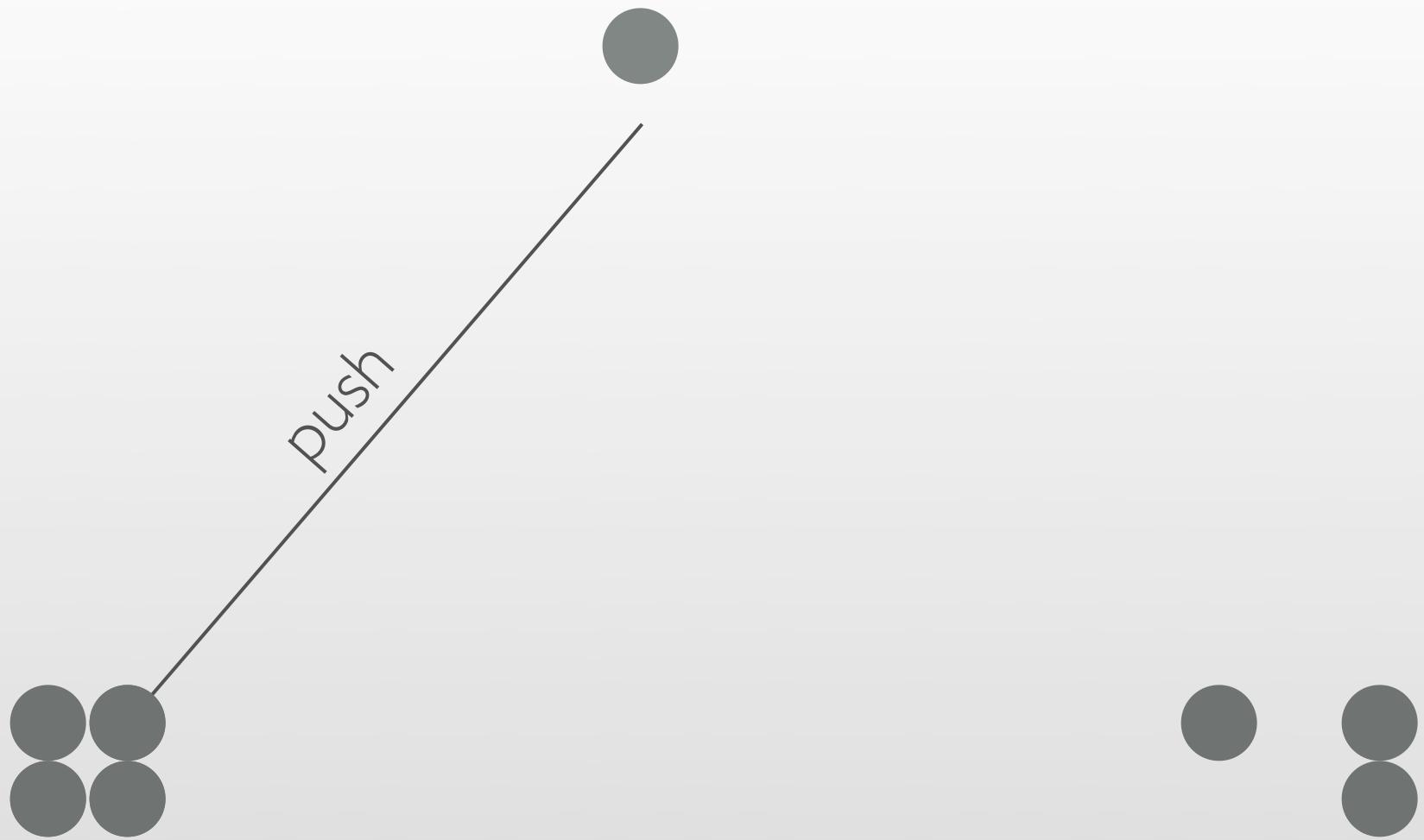
commit
commit
commit

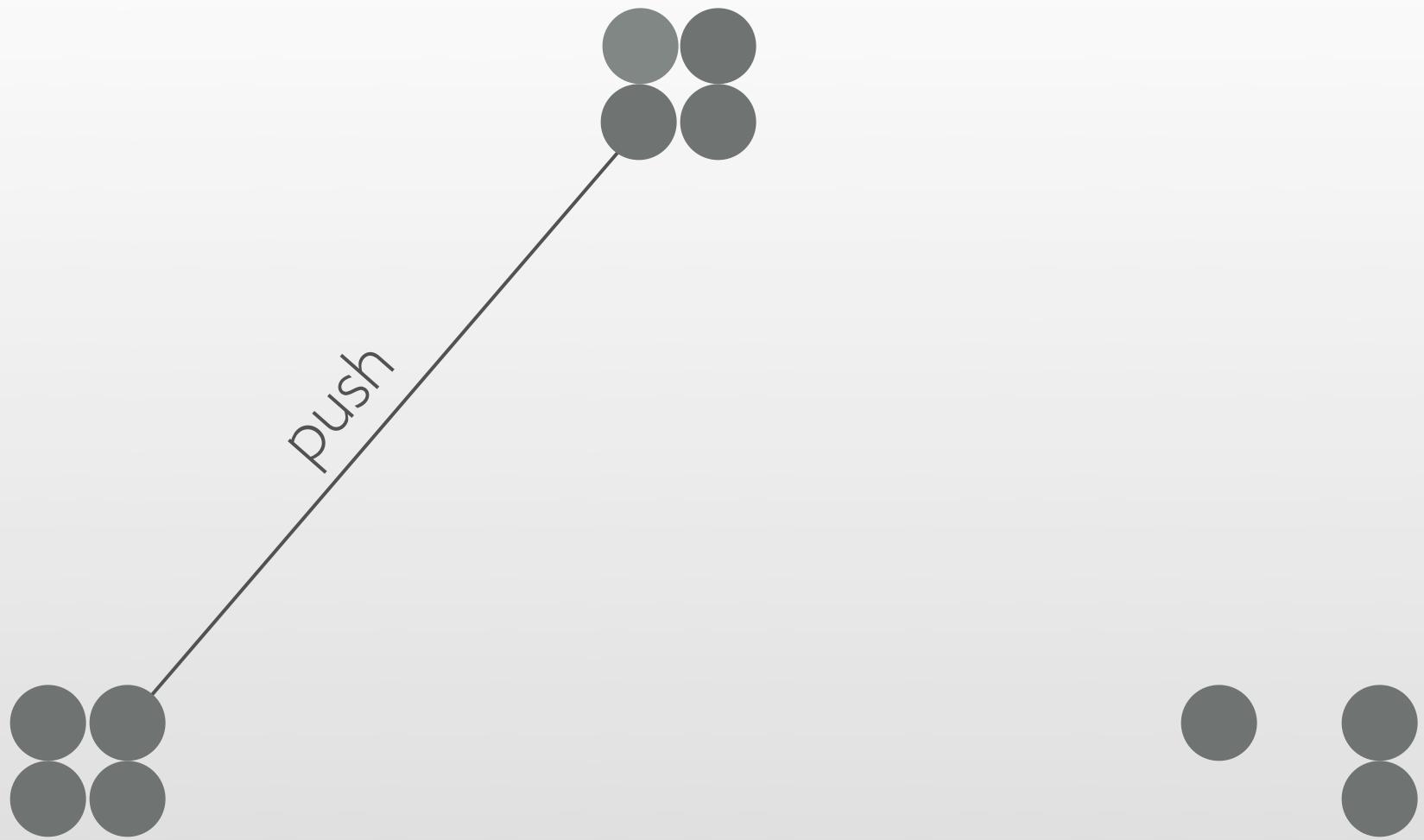


commit
commit
commit



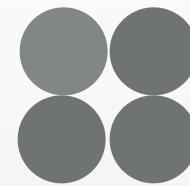


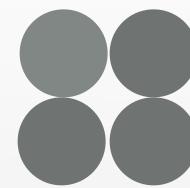


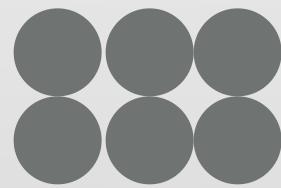
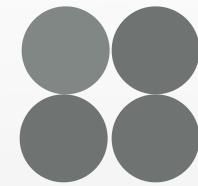




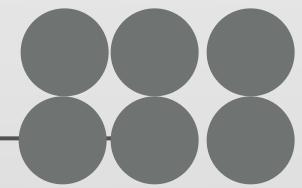
push

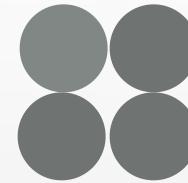




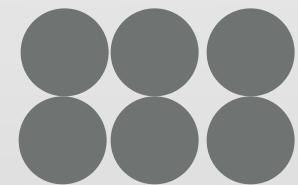
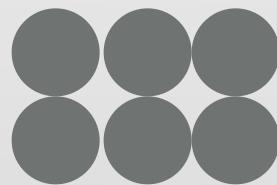


pull





Distributed



Location, Location, Location



Three Stage Thinking



Stash



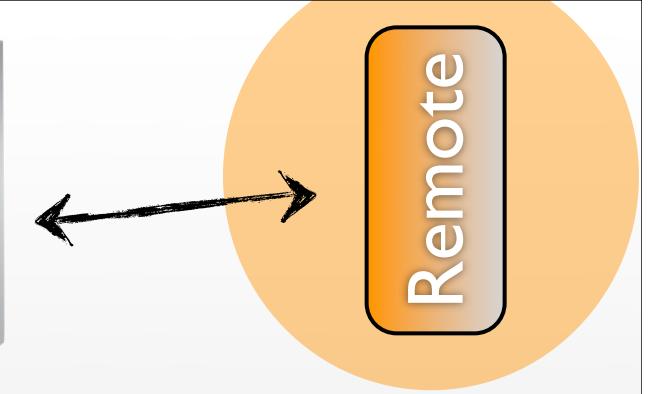
Working



Repo



Index



Remote

git clone

git checkout

edit project files

*boss interrupts
git stash*

*test bug on master
working as designed*

git stash apply

git add

git commit

git push

```
git clone git://somedomain/myproj.git ◀Build repo  
cd myproj.git  
  
git checkout master ◀Update index  
  
<Make algorithm code change in ClassOne.java> ◀Edit working copy  
  
git stash ◀Save to stash  
//Test the potential bug sans the algorithm change.  
//Old code was working as designed.  
  
git stash apply ◀Merge from stash  
  
git add ClassOne.java ◀Add to index  
git commit -m'Added comments' ◀Save to repo  
git push ◀Push to remote
```

Why?

Centralized version control systems

Folders

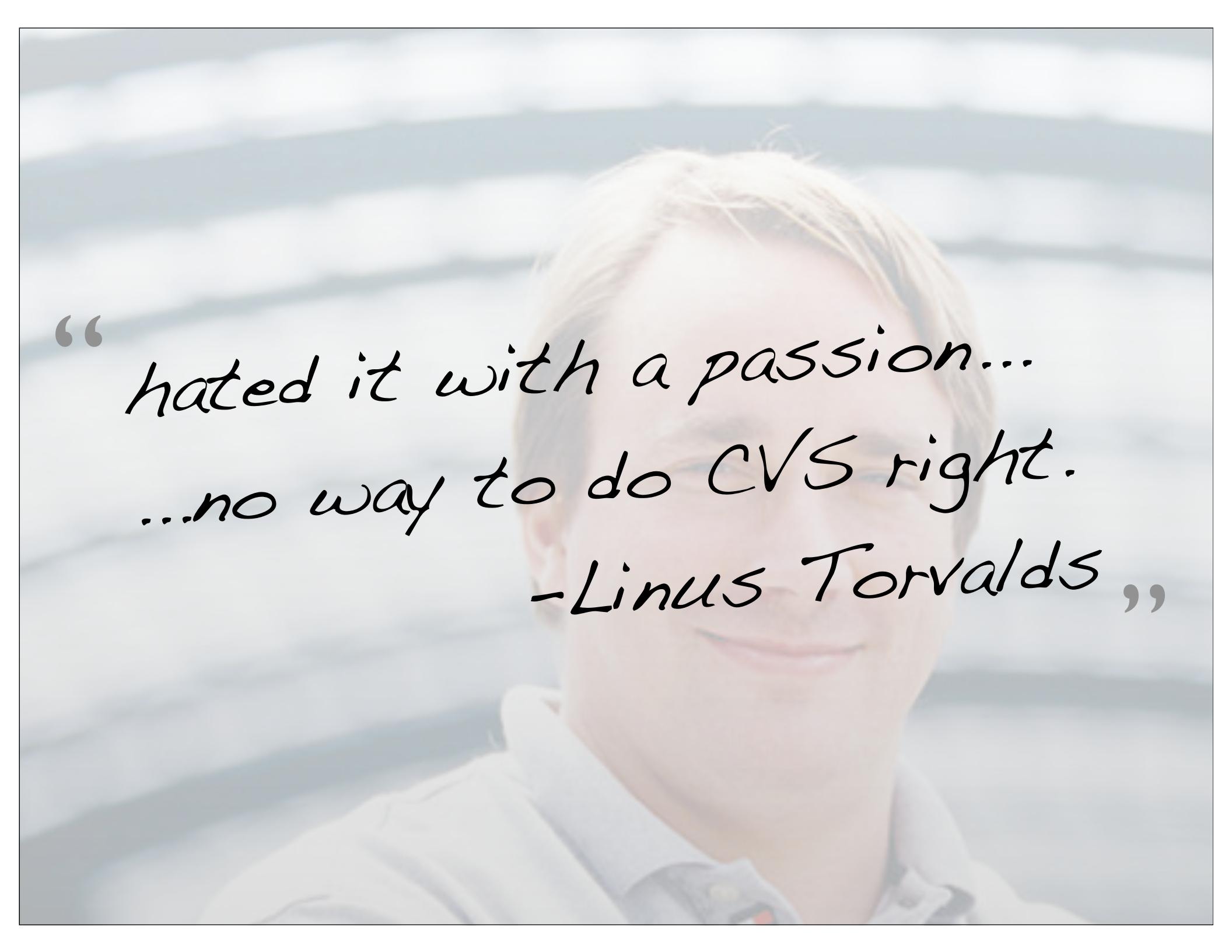
RCS

PVCS
Source
Safe

CVS
Clear
Case

Subversion

Perforce



“ hated it with a passion...
...no way to do CVS right.
-Linus Torvalds ,”

“ I did end up using CVS for 7 years at a commercial company and I hated it with a passion...

The slogan of Subversion for a while was "CVS done right"... and if you start with that kind of slogan, there's nowhere you can go.

There is no way to do CVS right.”

-Linus Torvalds

Distributed Version Control

A new idea?

1997

code co-op

200 |

arch

2003

monotone

2003

SVK

2003

darcs

2005

bazaar

2005

mercurial

2005

git

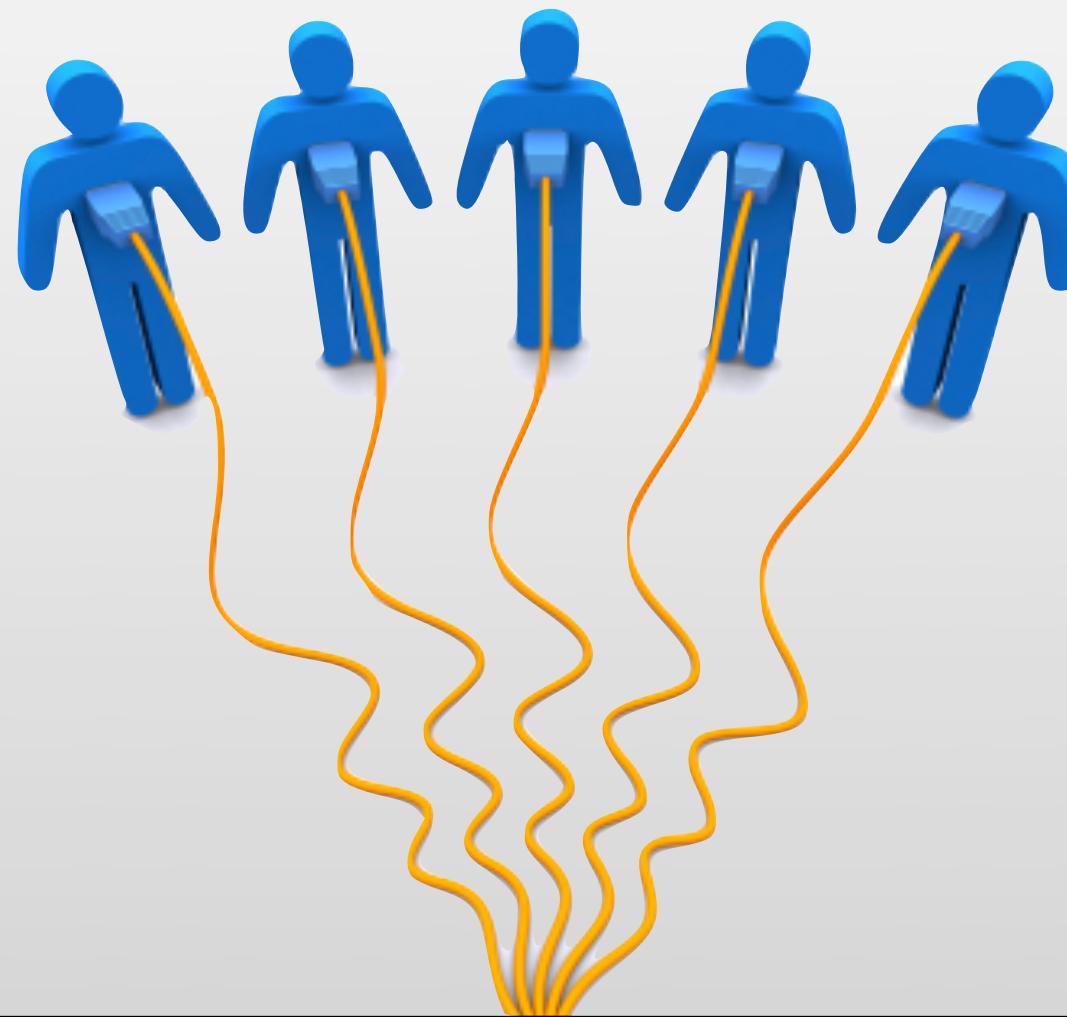
bazaar

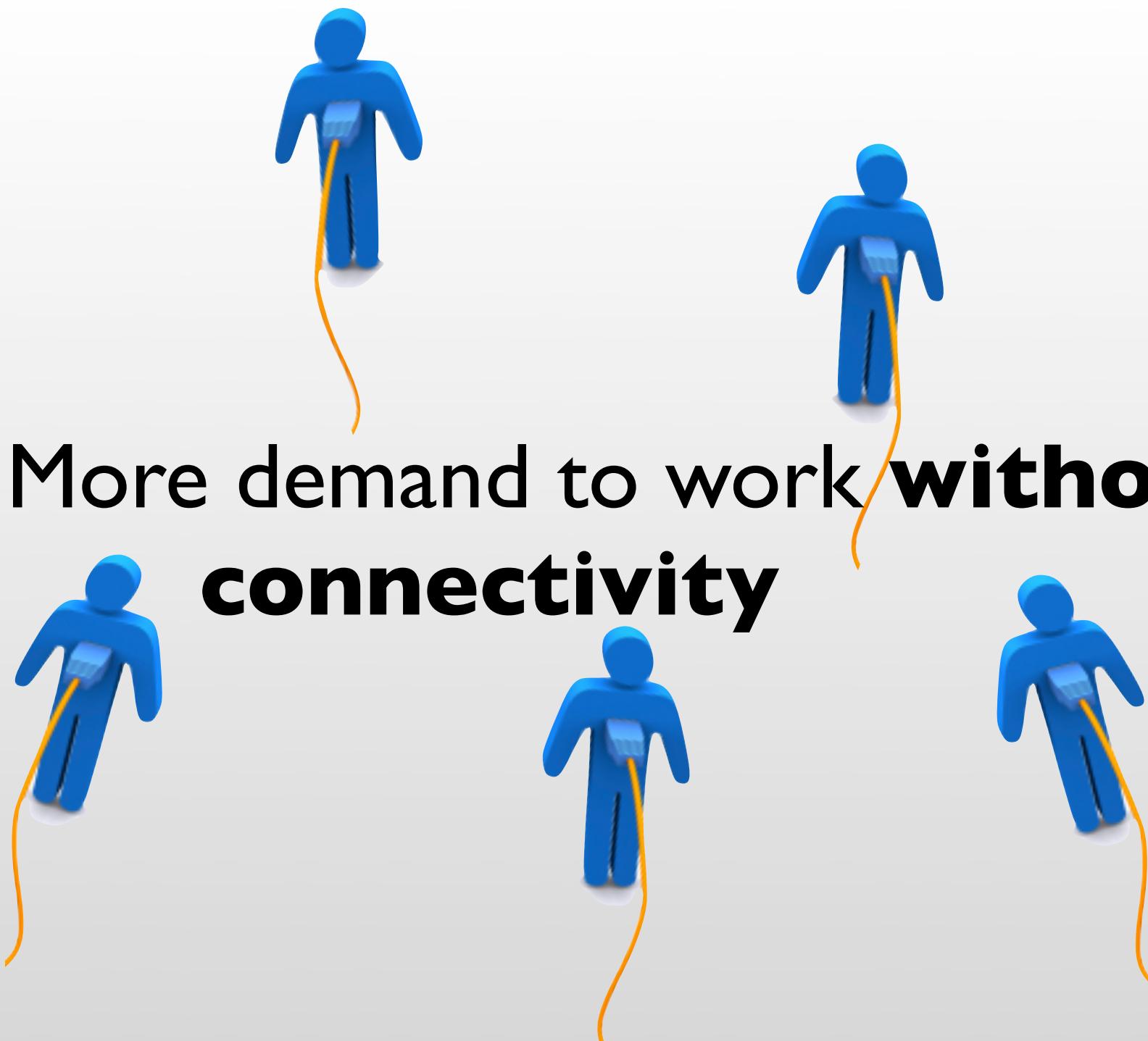
mercurial

git

Working Offline

More available **connectivity**





More demand to work **without**
connectivity

Offline Anything

Checkin

Add

Branch

List change log

Grep history

Rewrite history

Stash

Merge

Label

Remove

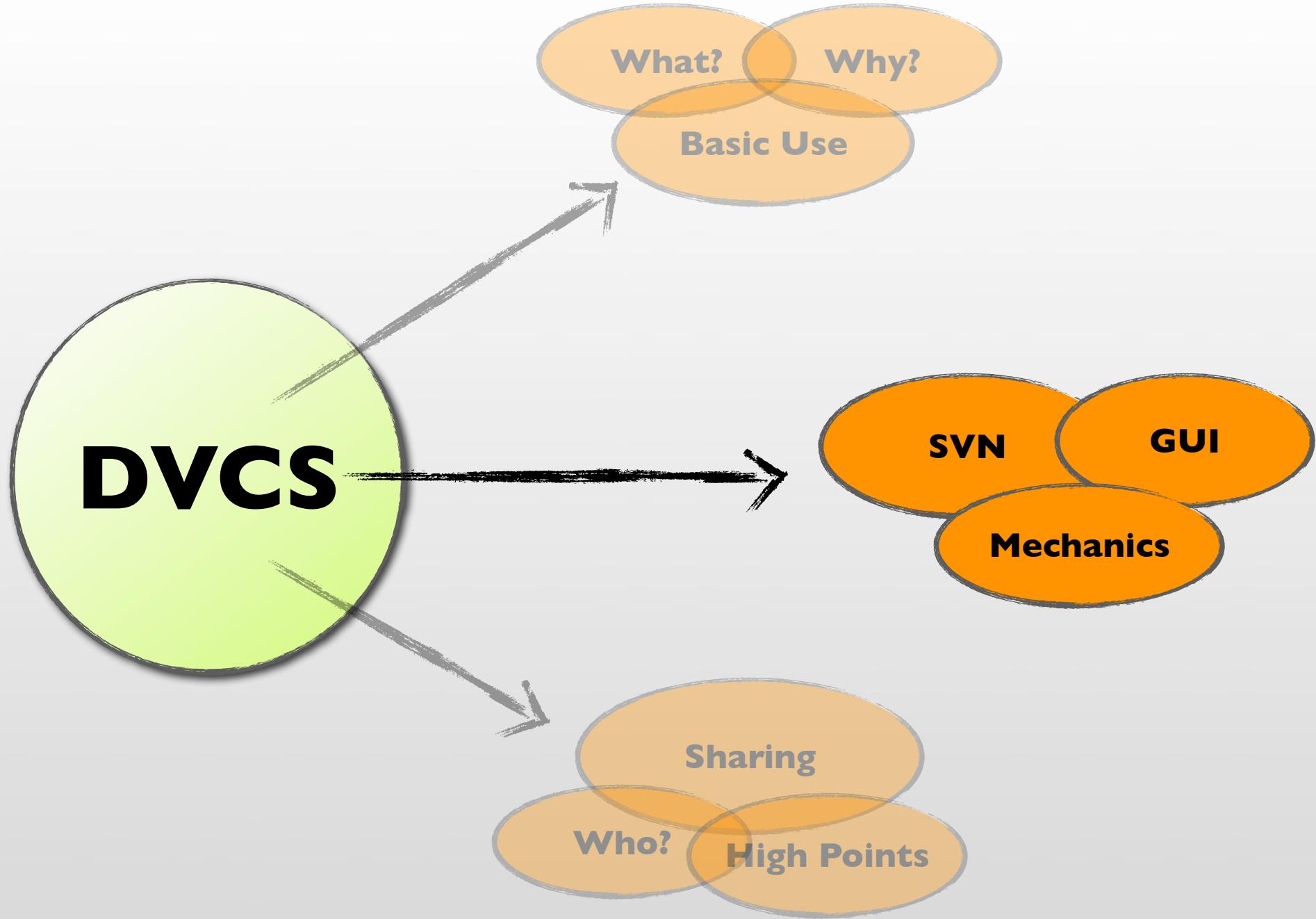
~~practically everything but push~~



A tropical beach scene featuring palm trees, a clear blue sky, and two dark wooden lounge chairs with red and white striped pillows on a sandy beach. A large orange speech bubble is positioned in the upper right, and a smaller red speech bubble is in the lower left, both containing text.

Matthew: Cool!
Another commit to my
latest OSS project using
Git!

Madelaine: I like
subversion
much better...



GUIs



GitK

gitk: sample11-gitscm.git

The screenshot shows the GitK application interface. On the left, a tree view displays the commit history of the 'sample11-gitscm' repository. The commits are color-coded: 'master' is green, 'remotes/origin/master' is blue, and other branches/commits are yellow. The commit list on the right shows the author (Matthew McCullough), email, date, and time for each commit. Below the commit list is a search bar with fields for SHA1 ID, Find, next, prev, commit, containing, Exact, All fields, and a search input field. On the right side, there is a 'Comments' section for 'pom.xml' and a 'Patch' section showing a diff view of the pom.xml file. The patch view highlights changes in the XML code.

SHA1 ID: 79f4a7be486d8c043c702bcd7f16e99ef2e56e42

Find next prev commit containing: Exact All fields

Search

Diff Old version New version Lines of context: 3 space

Author: Matthew McCullough <matthewm@ambientideas.com> 2009-03-01 17:35:38
Committer: Matthew McCullough <matthewm@ambientideas.com> 2009-03-01 17:35:38
Parent: dd25b10462942806bd562f50539934161737e72a ([maven-release-plugin])
Branches: master, remotes/origin/master
Follows: sample-git-scm-1.3
Precedes:

Merged in testwithoutplexus branch

----- pom.xml -----

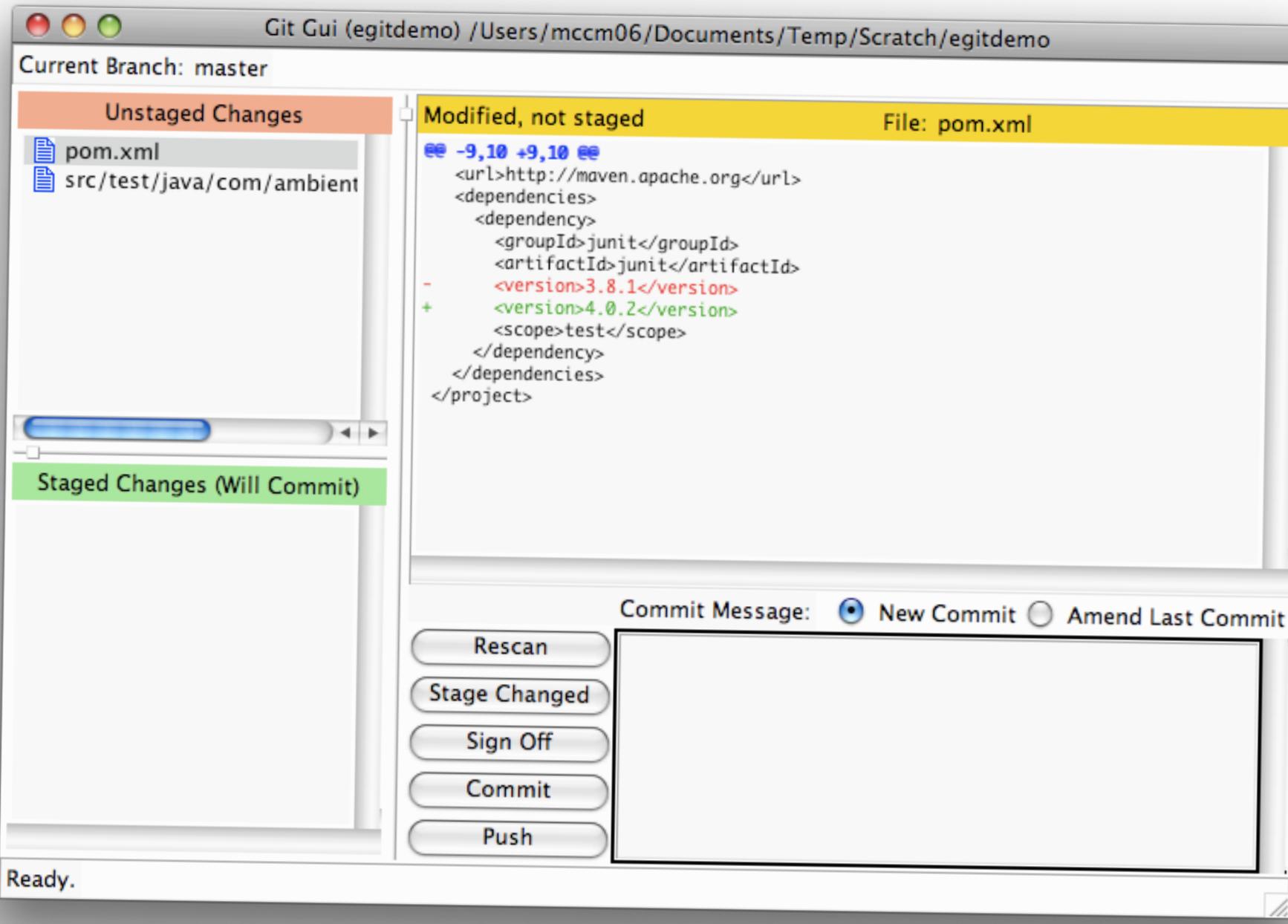
```
index f6c6391..2b4b01a 100644
@@ -4,7 +4,7 @@
<groupId>com.ambientideas</groupId>
<artifactId>sample-git-scm</artifactId>
<packaging>jar</packaging>
- <version>1.4-SNAPSHOT</version>
+ <version>1.7-SNAPSHOT</version>
<name>Sample Project - Git SCM</name>
<url>http://github.com/matthewmccullough/maven-training/tree/master</url>
```

```
@@ -20,30 +20,41 @@
<!-- <connection>scm:git:ssh://github.com/matthewmccullough/maven-training</connection>
<url>http://github.com/matthewmccullough/maven-training</url>
</scm>
```

Patch Tree

Comments
pom.xml

GitGUI

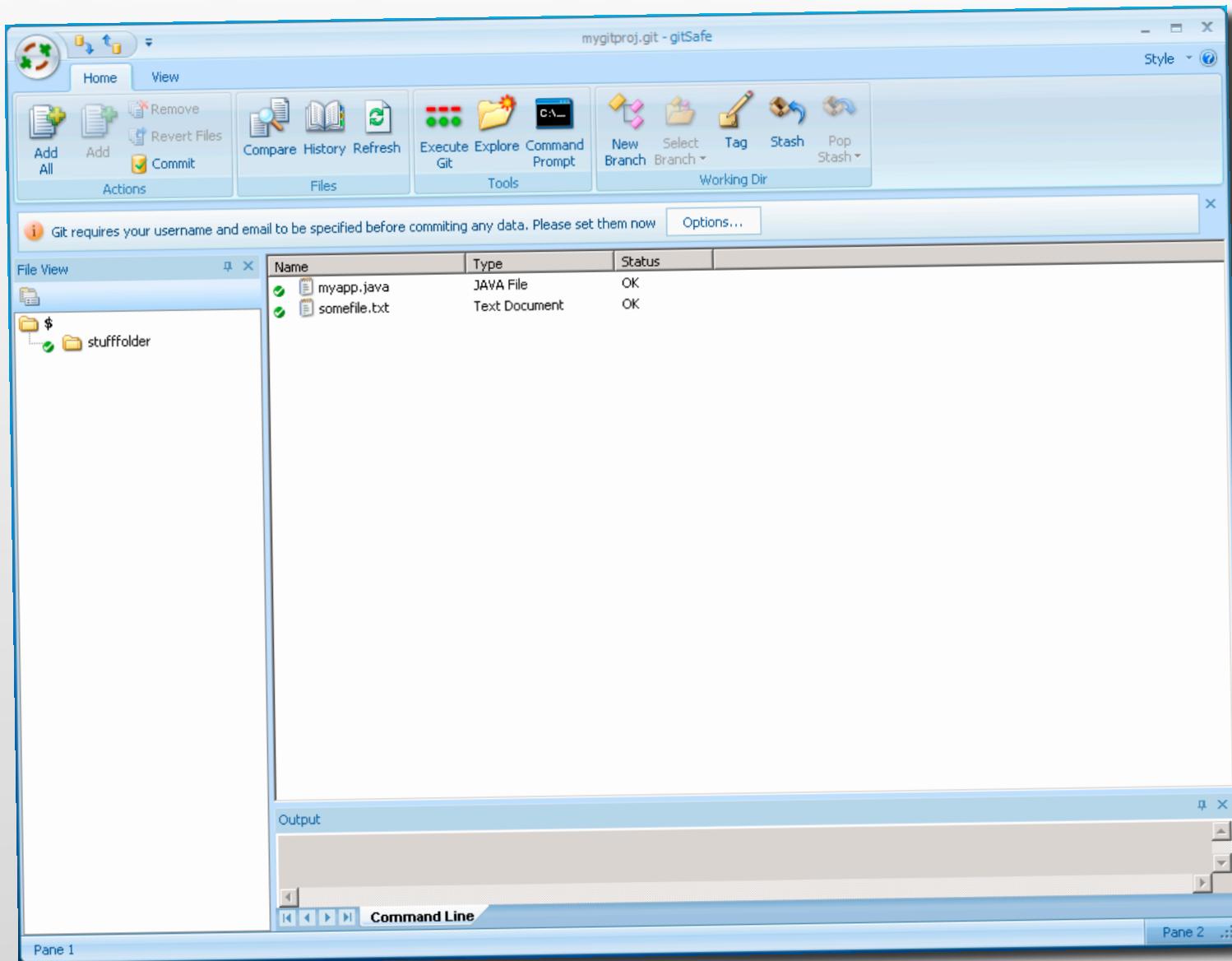


TIG

Terminal — tig — bash — 120x30

```
2009-07-31 10:17 Matthew McCullough + [master] [origin/master] [maven-release-plugin] prepare for next development iteration
2009-07-31 10:17 Matthew McCullough * [maven-cli-plugin-parent-0.6.9] [maven-release-plugin] prepare release maven-cli-
2009-07-31 15:58 shaolang M [shaolang/master] Merge branch 'master' of http://github.com/mrdon/maven-cli-plug-
2009-07-31 14:31 Don Brown |`-* Fixing broken build due to java 6 annotation
2009-07-31 15:46 shaolang *| Fixed bug in execute-phase where it ignored executing plugin goal.
2009-07-28 09:19 Matthew McCullough *| [maven-release-plugin] prepare for next development iteration
2009-07-28 09:19 Matthew McCullough *| [maven-cli-plugin-parent-0.6.8] [maven-release-plugin] prepare release maven-cli-
2009-07-28 08:38 Matthew McCullough M Merge commit 'shaolang/master'
2009-07-27 21:08 shaolang |`-* Renamed object member commandCallBuilder to phaseCallBuilder.
2009-07-27 16:30 shaolang M Merged jroper/master, and fixed bug (introduced) where execute-phase ignores us-
2009-07-27 21:32 Matthew McCullough *| | [matthew/HEAD] [matthew/master] [master-matthew] [maven-release-plugin] prepar-
2009-07-27 21:32 Matthew McCullough *| | [maven-cli-plugin-parent-0.6.7] [maven-release-plugin] prepare release maven-
2009-07-27 21:23 Matthew McCullough *| | Implemented JROPER's fixes that he did to CommandCallBuilder into PhaseCallBu-
2009-07-27 15:34 Matthew McCullough M | Merged in Shaolang's refactorings, but this erased Jroper's fix, so I'll have-
2009-07-23 20:14 shaolang |`-* Changed CliConsoleCommandDescription to automatically print the token descrip-
2009-07-23 20:10 shaolang * Removed the need to make phasesAndProperties in ExecutePhaseCommand unmodifiabl-
2009-07-23 20:02 shaolang * Replace the dependency of Log to CliConsole in PhaseCallRunner, and added in th-
2009-07-23 19:39 shaolang * Moved the creation of userAliases, commands and prompt to the declaration, inst-
2009-07-23 19:18 shaolang * Removed CliShell's dependency to implement interface Runnable.
2009-07-23 19:08 shaolang * Made CliConsoleCommandDescription to print a new-line just before printing the ~
2009-07-23 17:47 shaolang * Made CliConsoleDescription#formatTokenDescriptions() to left-align the printing-
2009-07-23 14:15 shaolang * Added maven-cli-plugin to pom.xml.
2009-07-23 13:57 shaolang * Removed unused imports in IdeaMojo.
2009-07-23 13:37 shaolang * Reduced visibility of AbstractCliMojo#displayShell() to private.
2009-07-23 13:34 shaolang * Moved code from AbstractCliMojo#buildValidCommandTokens() to AbstractCliMojo#bu-
2009-07-23 11:34 shaolang * Changed AbstractCliMojo to create CommandsCompleter only once, instead of every-
2009-07-22 19:28 shaolang * Fixed the bug (introduced) where cli will terminate when it cannot successfully-
2009-07-22 15:53 shaolang * Pulled codes from ExecuteCliMojo to AbstractCliMojo, thereby making ExecuteClip-
[main] 1d090ed2731274e7d2fbe999b5aa8717b1b13211 - commit 1 of 155 (18%)
```

GitSafe



GitX

hellogitworld.git

View Branch Create Branch

Subject

- master Added a new file
- Changes to text
- Small edit on master
- Small edit
- A fix by Matthew
- Fun with Tim
- Another file
- Merge branch 'master' of git@github.com:matthewmccullough/helloworld
- First checkin
- Removing conflicting file

```
commit 4e0d383949974d9bd6ff977900bb991794b717ba
tree 1b0cf0e780b227472cb8c89697e7f4429f34d531
parent 0bbe90b062b4a75f7d0c87903a232c9a058b2f2b
author Matthew McCullough <matthewm@ambientideas.com> 1231997293 -0700
committer Matthew McCullough <matthewm@ambientideas.com> 1231997293 -0700

Added a new file

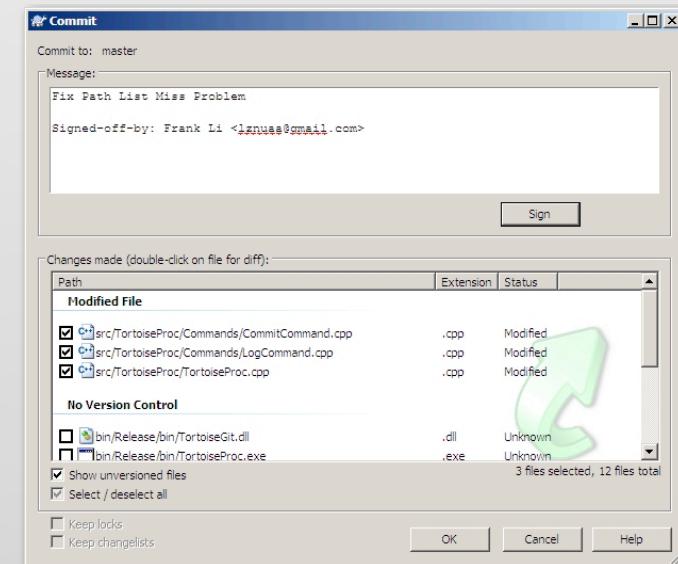
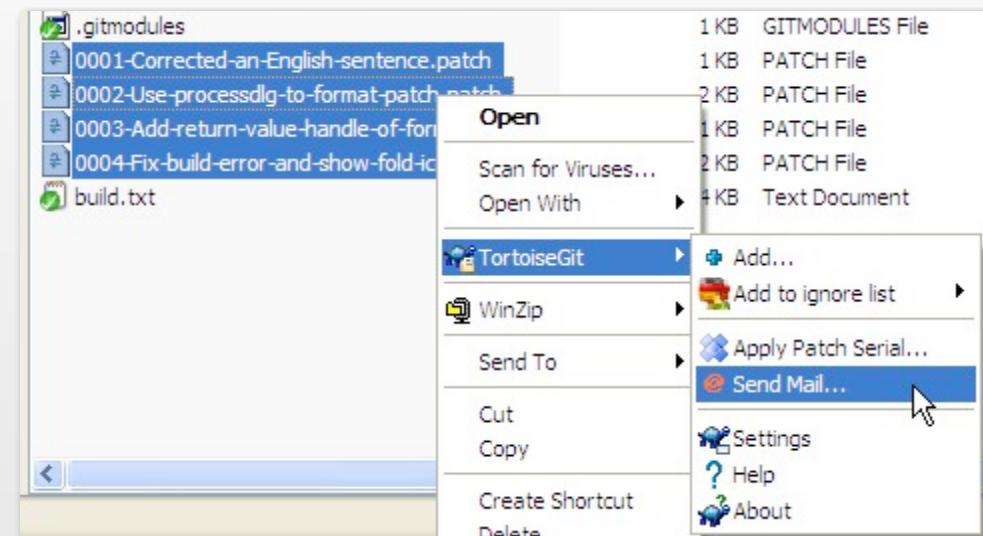
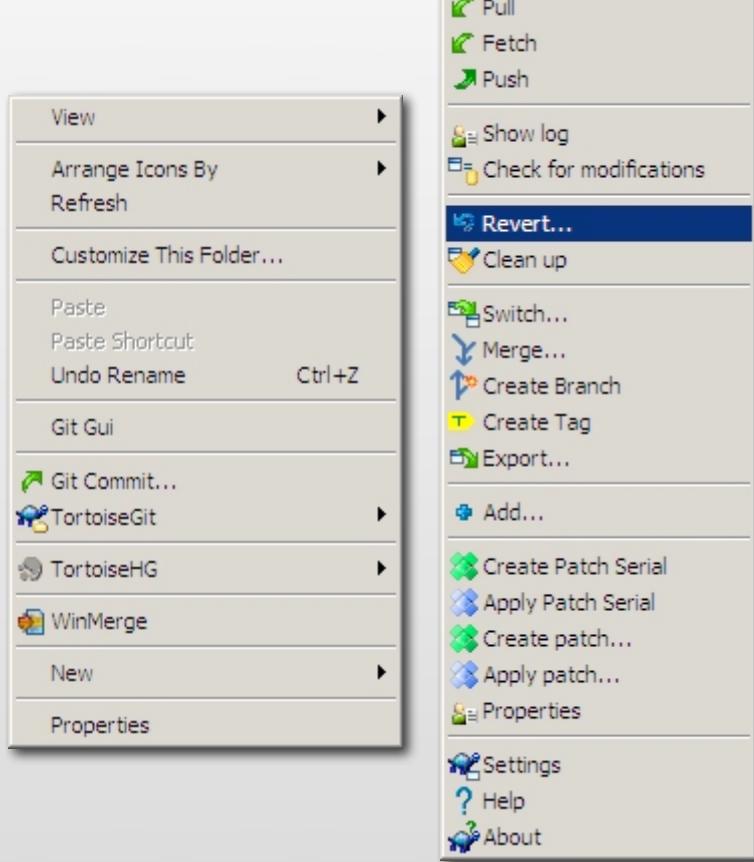
diff --git a/anewfile.txt b/anewfile.txt
new file mode 100644
index 000000..e69de29
```

Author Date

Author	Date
Matthew McCullough	2009-01-14 22:28:13
Matthew McCullough	2009-01-14 22:24:03
Matthew McCullough	2008-12-28 21:21:29
Matthew McCullough	2008-12-28 21:11:31
matthewmccullough	2008-12-14 19:17:11
Matthew McCullough	2008-11-25 19:49:36
Matthew McCullough	2008-11-24 21:42:29
Matthew McCullough	2008-11-24 21:41:29
Matthew McCullough	2008-11-24 21:22:01
Matthew McCullough	2008-11-24 21:10:41

11 commits loaded

TORTISEGIT



eclipse.org page

http://www.eclipse.org/proposals/egit/ Google

CONTACT | LEGAL

eclipse

HOME | USERS | MEMBERS | COMMITTERS | DOWNLOADS | RESOURCES | PROJECTS | ABOUT US

Links ▾

> Proposals

Eclipse Git Team Provider (EGit)

Proposal posted 3 hours ago

Introduction

The EGit project is a proposed open source project under the Eclipse Technology Project.

According to the Eclipse Development Process, this proposal is intended to declare the intent and scope of the EGit project as well as to gather input from the Eclipse community. You are invited to comment on and/or join the project. Please send all feedback to the <http://www.eclipse.org/newsportal/thread.php?group=eclipse.egit> newsgroup.

EGit is an Eclipse Team provider for [Git](#). Git is the version control system originally developed for the Linux kernel by Linus Torvalds. One of Git's strength is its ability to support distributed version tracking, as well as more centralized schemes, in contrast with traditional SCM tools that offer only a centralized model (such as CVS and Subversion). Git is also very fast. Its usage has expanded beyond the Linux kernel team and is now used by many other projects, both for open source and commercial development, as well as fast and distributed general purpose Source Code Management (SCM).

Git is a distributed SCM, which means every contributor has a full copy of the history of every revision of the code, making queries against the history very fast and versatile. The distributed model also means most operations are up to a hundred times faster than with traditional centralized SCM such as CVS or Subversion. The integrity of the code is guaranteed through the use of secure hashes on the content and history, plus optional GPG/PGP key signing of commits.

Background

Traditional source code control systems have been based around a single centralized server housing all of the project's revision history and assets. Historically this has worked well for smaller projects, especially when all project contributors have access to the same computer system and are located in the same physical workplace. Distributed open source projects (and commercial entities or virtual companies) have broken this model, making it much more difficult for contributors to participate. A centralized model promotes two classes of contributors: "blessed people" (those who have access to make changes to the project and use the source code control system) and "everyone else" (the rest of the world). Moving between classes is painful for everyone involved, and in many cases moving from "everyone else" to "blessed people" requires months of proving the contributor is trustworthy enough.

Multiple distributed version control systems (DVCS) have recently developed to fully support the truly distributed nature of modern open source projects, in particular large scale efforts such as those supported by Eclipse, Mozilla, KDE, Open Solaris, and the Linux Kernel. Git, Mercurial, Monotone, Bazaar and darc are all popular implementations of the DVCS concept, however their implementation details differ significantly.

True to their name, DVCS implementations eliminate the centralized server and place all project participants on an equal footing by providing them with identical capabilities. Brand new project contributors have access to the same tools and historical data as long-term trusted contributors, making it possible for new contributors to participate on an equal footing with no risk to the overall project. Most projects using DVCS tools flow changes from newer contributors to trusted contributors, arranging a loose hierarchy of review.

DVCS tools tend to perform better than their centralized counterparts, as network latency is only a factor when exchanging batches of new revisions. In a DVCS the most commonly used operations such as recording changes, file comparison, history inspection, branch creation and branch merging are handled entirely offline using locally stored information, allowing these operations to complete as quickly as the disk buffer cache and CPU will permit. Offline operation can be very useful in distributed open source projects where not all participants have large network bandwidth with low latency to connect them to the project's central server(s). For some participants, being able to work offline may be a necessity for them to contribute, due to limited availability (or high cost) of Internet access.

Finally, the implications of adopting a distributed SCM approach can bring significant benefits, especially for large open source projects such as Eclipse, and can help lower the barrier to entry for new contributors and make source code experimentation easier.

ITS Official!



Visit other Eclipse Sites



Home Download Documentation Support Developers

Google™ Custom Search



EGit

 **Download**
Eclipse Distribution,
Update Site, Dropins



Documentation

Tutorials, Examples,
Videos, Online Reference



Documentation

EGit

EGit is an Eclipse Team provider for the [Git version control system](#). Git is a distributed SCM, which means every developer has a full copy of all history of every revision of the code, making queries against the history very fast and versatile.

The EGit project is implementing Eclipse tooling on top of the [JGit](#) Java implementation of Git.

Incubation

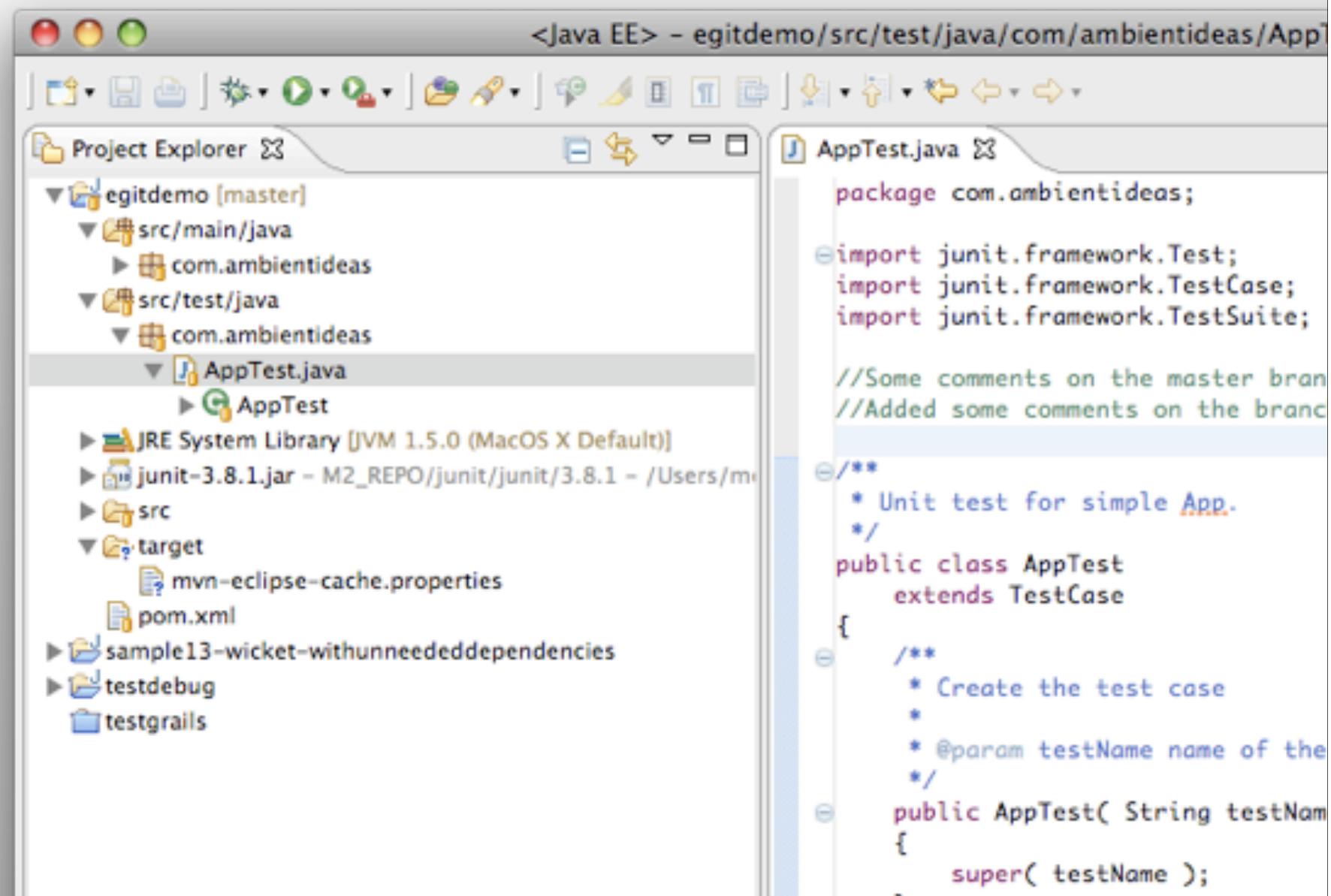
EGit is in the [Incubation](#) P


<http://delicious.com/matthew.mccullough/git>

Committers

- Chris Aniszczyk (Red Hat)
- Christian Halstrick (SAP)

Egit



<Memory An... <Debug <Java EE>

Help History X

egitdemo

	Author	Date
master HEAD Merged	Matthew McCullough	2009-04-03 09:29:43
Master branch comments	Matthew McCullough	2009-04-03 09:29:43
featurecard... Some branch	Matthew McCullough	2009-04-03 09:29:43
First checkin	Matthew McCullough	2009-04-03 09:29:43

commit
3554e29cdf877f7922894b88fc38349d8900e33
Author: Matthew McCullough
<matthewm@ambientideas.com> 2009-04-03
09:29:43
Committer: Matthew McCullough
<matthewm@ambientideas.com> 2009-04-03
09:29:43
Parent:
[19b3a479510ca07dce93a3f1e59275ebe734dab7](#)
(Master branch comments)
Parent:
[5217e55489ebad0a6c620fc28bf25d8a5ec687e4](#)

Path
src/test/java/com

```
;;
:Case;
:Suite;

ter branch
ie branch for the egit demo
```

J-

```
e of the test case
```

```
testName )
```

```
F tests being tested
```

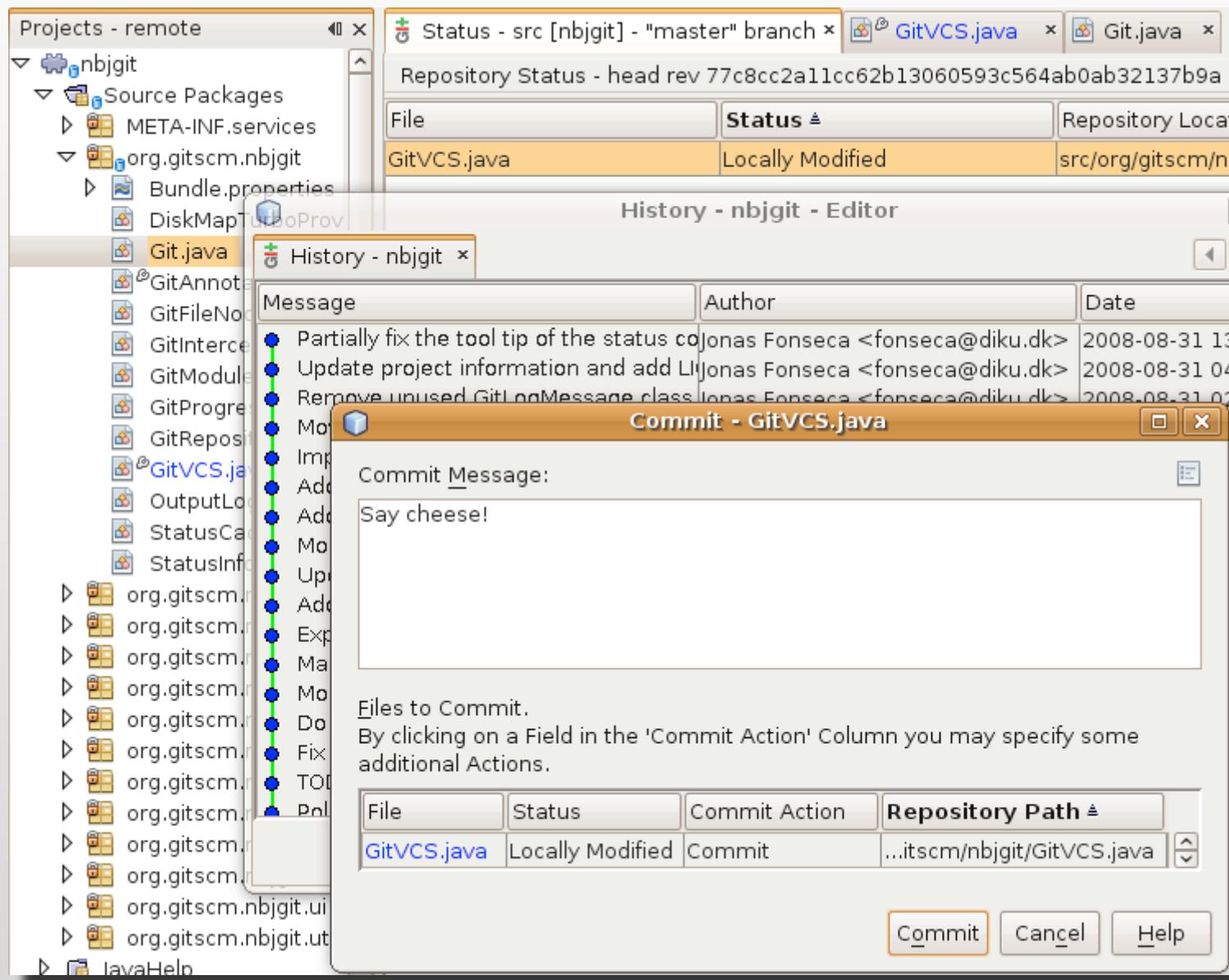
```
te()
```

```
te( AppTest.class );
```

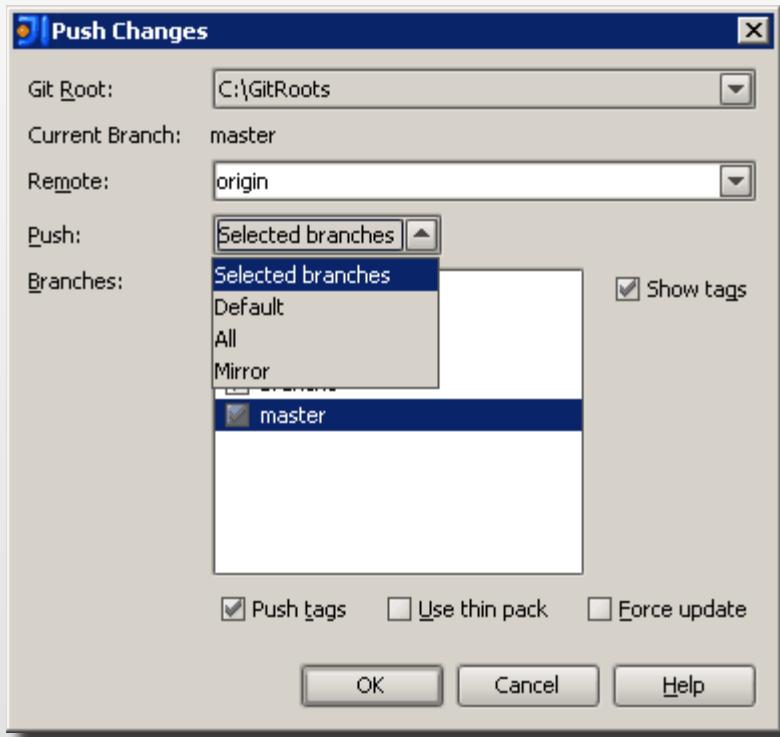
```
erities
```

Resource	Path	Location
----------	------	----------

NETBEANS



IntelliJ



The Rebasing commits dialog box displays a commit list:

Action	Commit	Comment
pick	215d838	Urgent fix
edit		
skip		
squash		

Buttons: View, Move Up, Move Down.

The Paths affected in commit 215d838 dialog box shows the file structure:

- Changed Files
- Idea-test-rebase2 (1 file)
 - C:\GitRoots\idea-test-rebase2\src (1 file)
 - rebase-test2-file1

HASHES AND Git



Index vs Hash

- ▶ Centralized VCS uses DB sequential index.

Index vs Hash

- ▶ Git uses SHA-1 hash.

40 hex characters

9AB223D28B1AA46EF1780B22F304982E39872C34

use as **little** of it as is **unique**

9AB223D28B1AA46EF1780B22F304982E39872C34

9AB223D28B1AA46EF1780B22F304982E39872C34

treeish

treeish = shorthand for hashes

9AB223

9AB223^

HEAD

HEAD[^]

HEAD~3

--since="2 weeks ago"

9AB223..56CD77

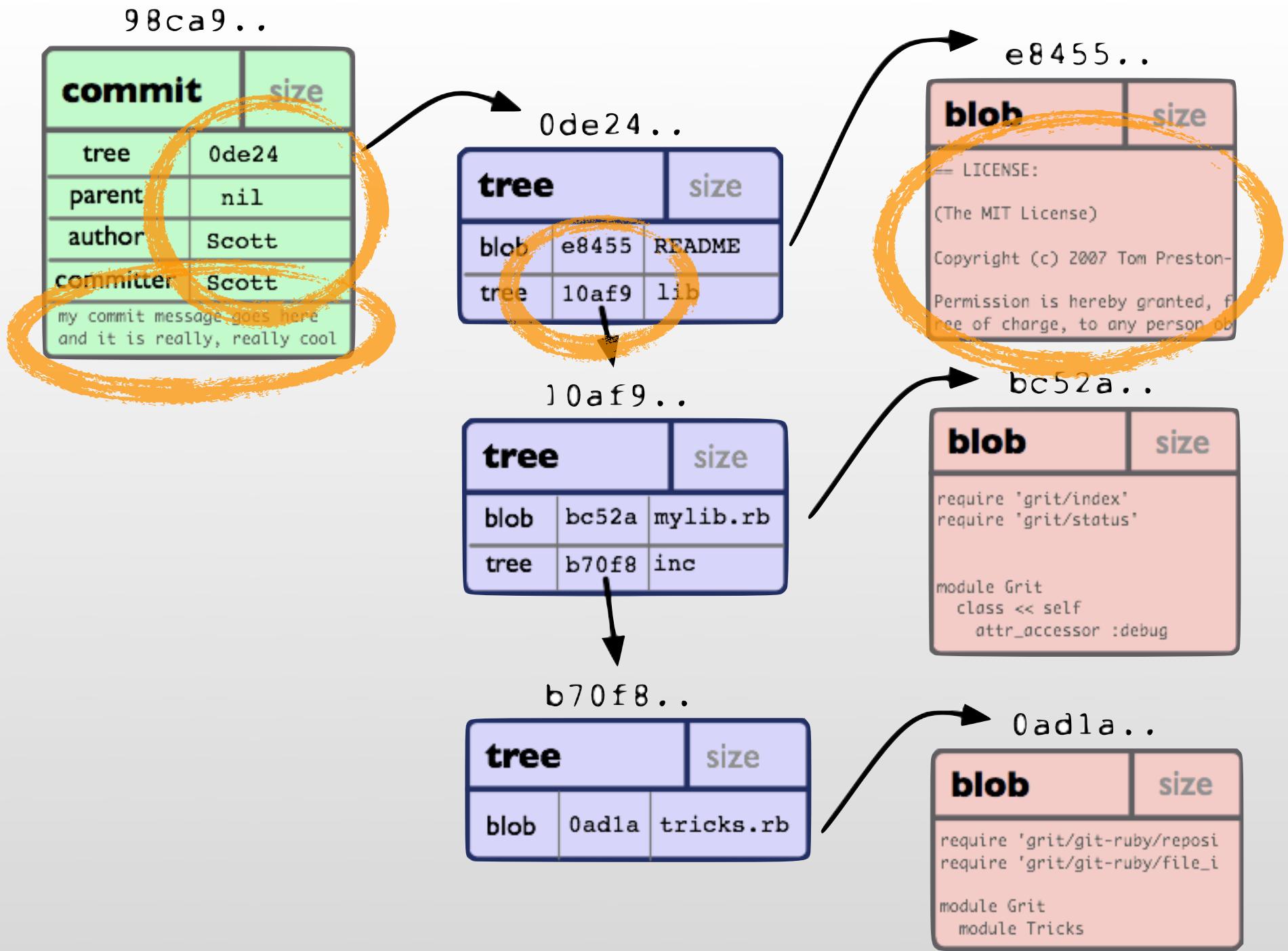
HASH BENEFITS

Independence

- ▶ Time & location separation from sequence # of checkin.
- ▶ Easy identification of what needs to be synced

HASHABLE Objects

- ▶ Blob
- ▶ Tree
- ▶ Commit
- ▶ Tag



HASH BENEFITS

Integrity

- ▶ Identifies damaged repos.
- ▶ Prevents modification of published history.
- ▶ Unique to file size and contents.

HASH BENEFITS

Tags

- ▶ Cryptographically sign tags by GPG.
- ▶ Guarantee repo's state at a point in time.

Storage Mechanics

Typical SCMs use **deltas**

cvs / Subversion / darcs / Mercurial

**Delta storage gets slower as the history
of a file gets longer**

**Git uses Directed Acyclic Graph storage
(DAG)**

```
cp -r srcfolder srcfolder.prev
```

Copy of the **entire tree** per checkin

You've got to be kidding me!

zlib deflates every blob at commit

symlinks to existing **identical** blobs

400MB Subversion repo
convert to a **70MB** Git repo



SPEED

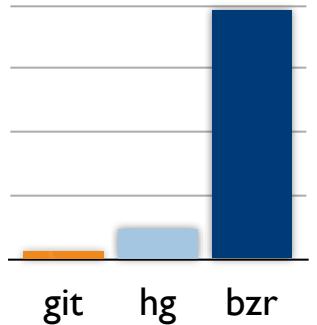
Init



Add



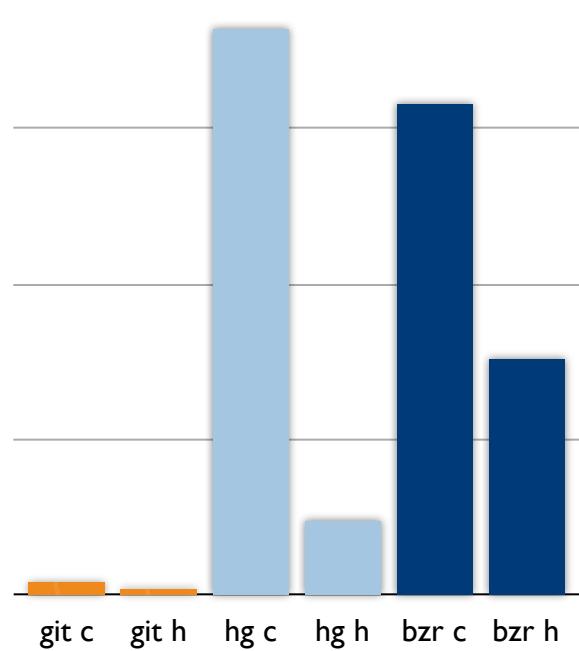
Status



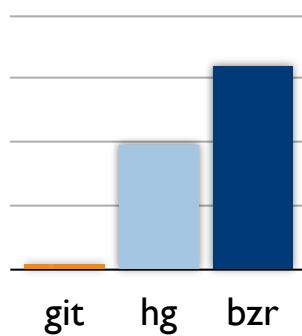
Diff



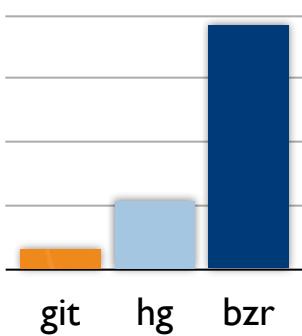
Branch (Cold/Hot)



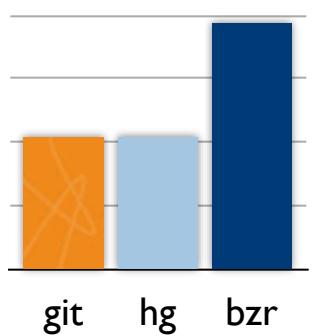
Tag



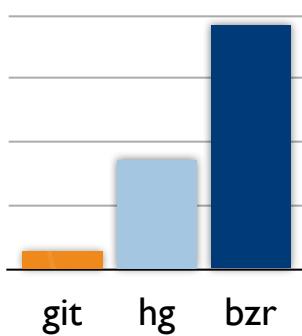
Log



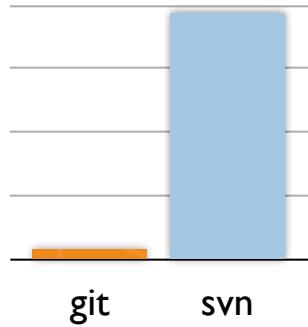
Commit (Lg)



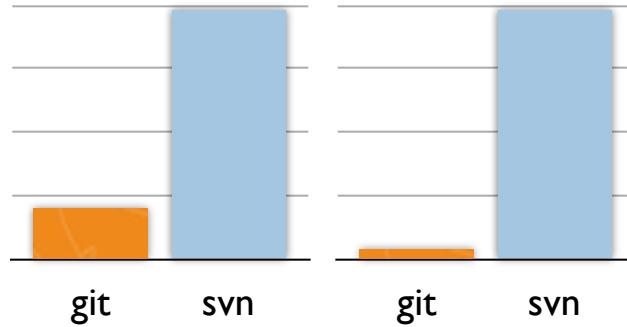
Commit (Sm)



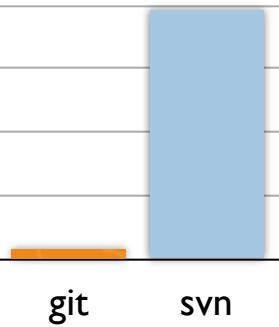
Init



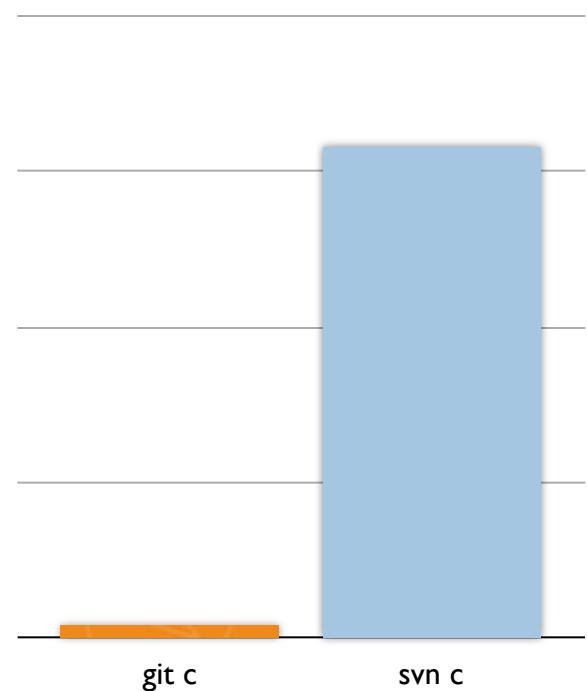
Status



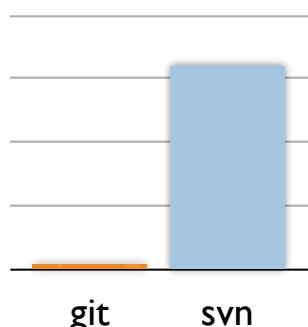
Diff



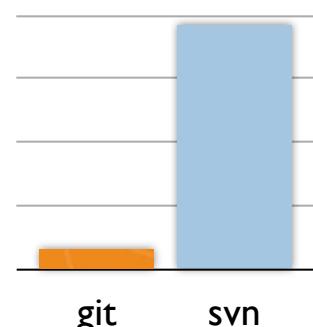
Branch



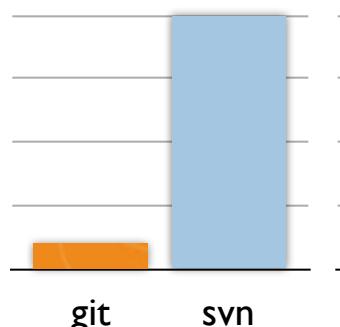
Tag



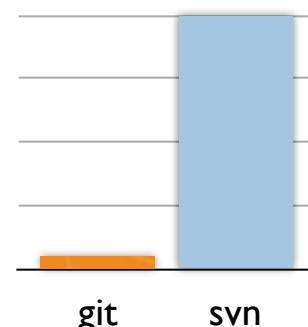
Log



Commit (Lg)



Commit (Sm)



10
TO

100 TIMES

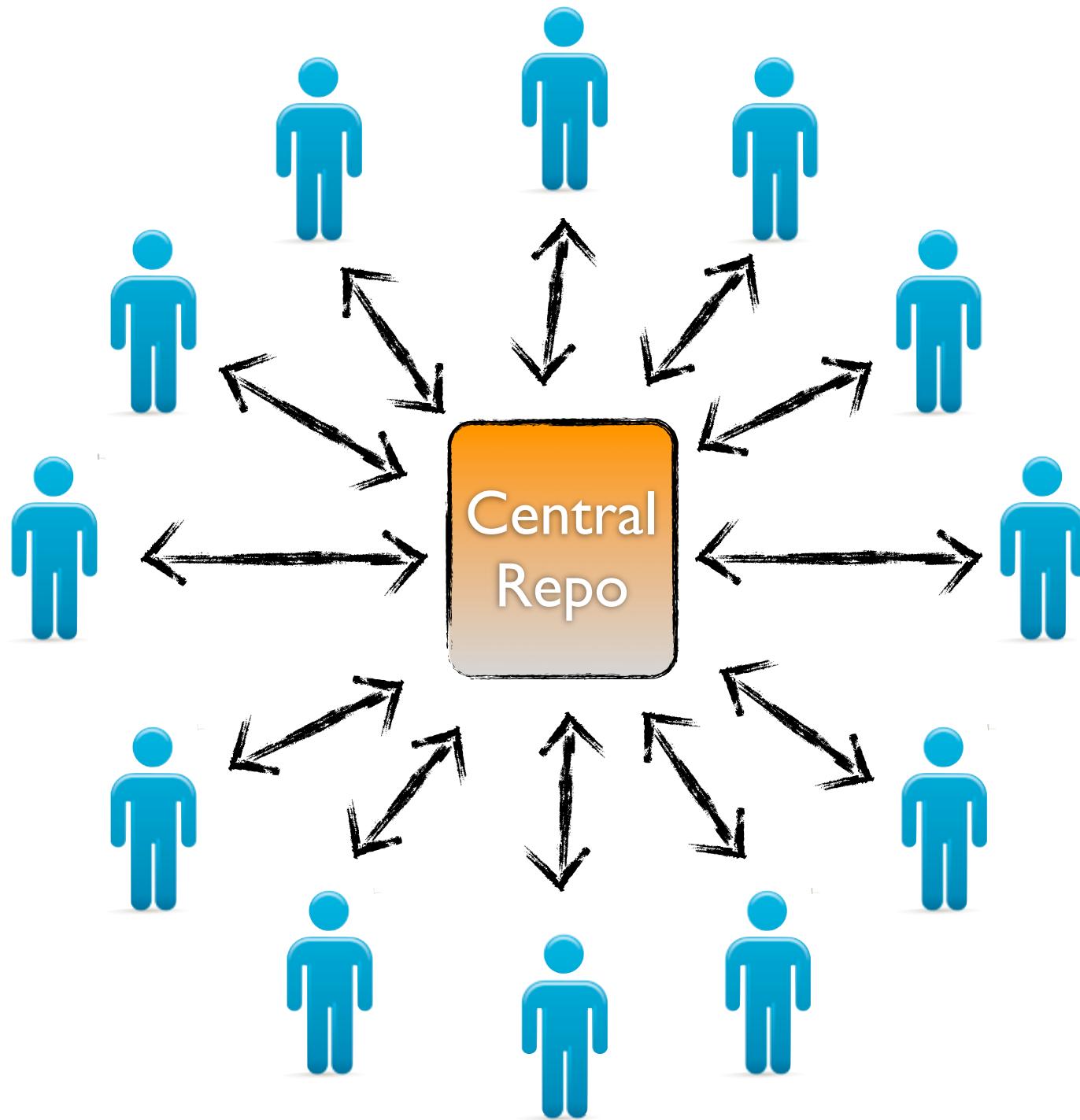
FASTER

<http://whygitisbetterthanx.com>

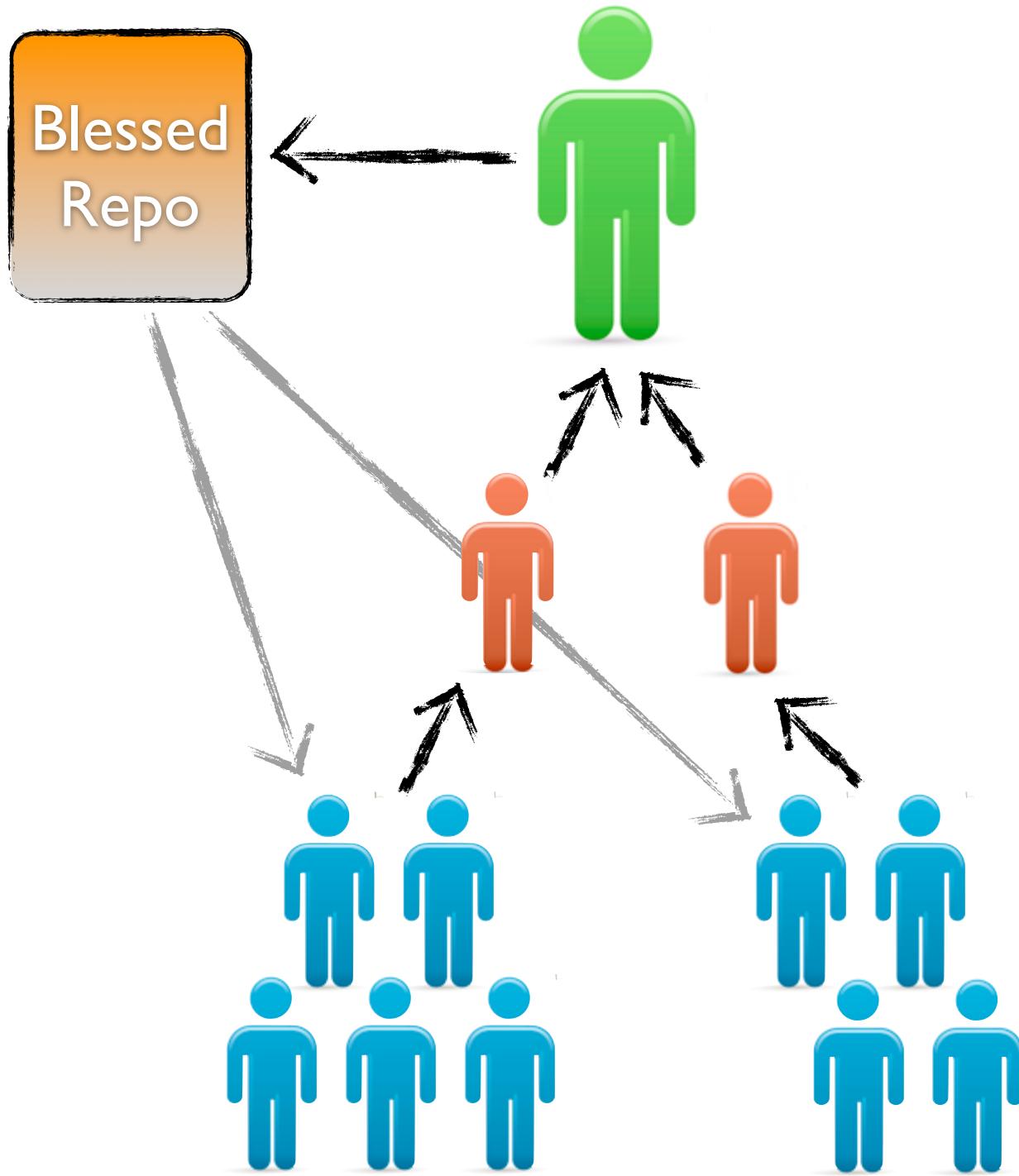
USAGE Models



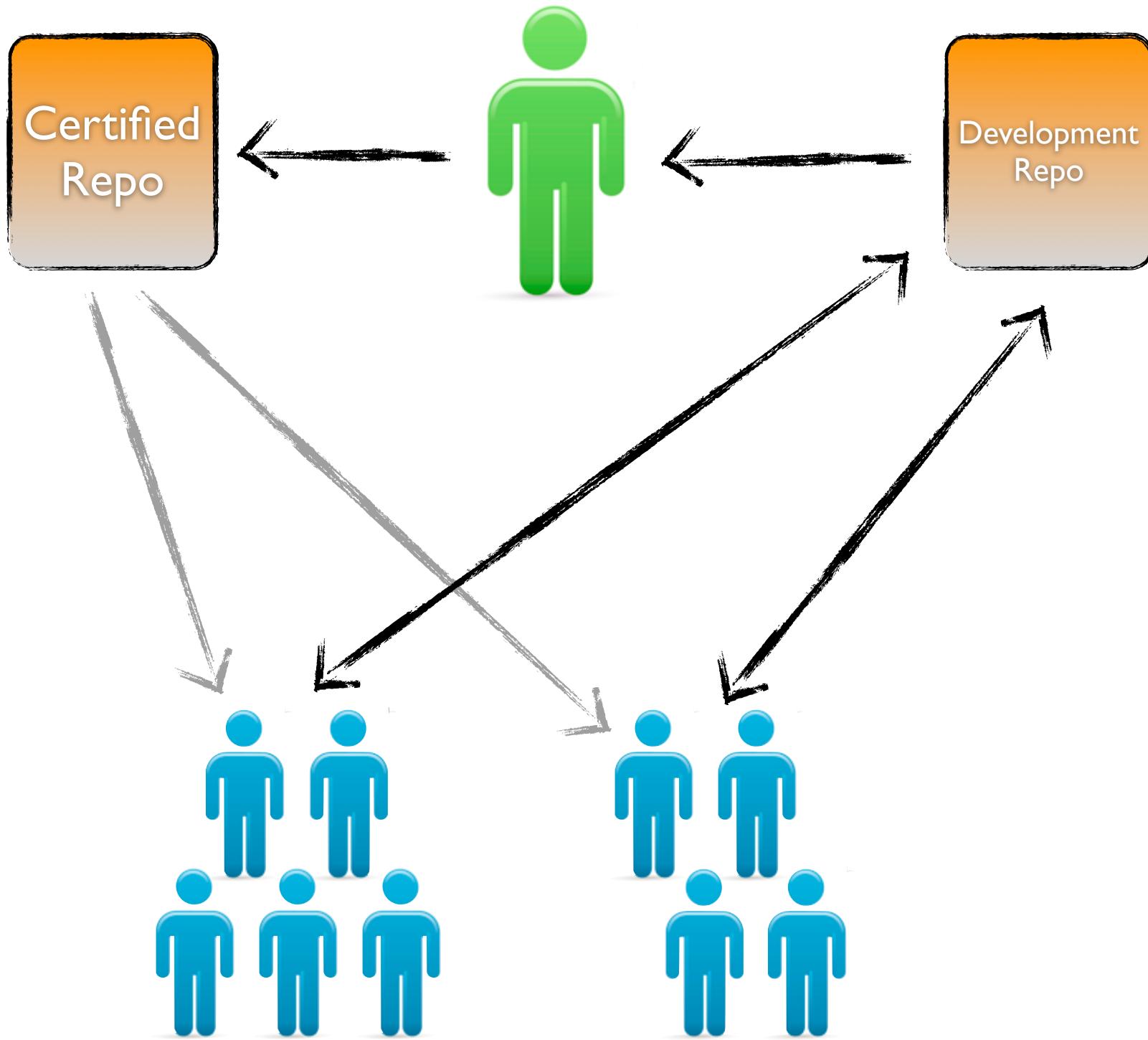
Centralized



Dictatorship



Integration Managed



SVN INTEROP



arguably the **best feature** of git

use git in **stealth mode**

SubVERSION

```
git svn clone http://unfurl.com/trunk  
//Hack, hack, hack  
git commit -a -m 'Some changes'  
git commit -a -m 'More changes'  
git commit -a -m 'Final changes'  
git svn dcommit
```

Ambient Code 1

Session

```
$git svn dcommit
Committing to http://ambientideas.unfuddle.com/svn/ambientideas_demo ..
.
A      anewfile2.txt
Committed r7
A      anewfile2.txt
r7 = 7772d0025962d976bcec34151155244c5af91153 (git-svn)
No changes between current HEAD and refs/remotes/git-svn
Resetting to the latest refs/remotes/git-svn
$
```

SubVERSION

- ▶ First class compatibility.
- ▶ Round trip support.
- ▶ Git commits = svn commits.

effortlessly convert full history
when ready to migrate the team

Migration

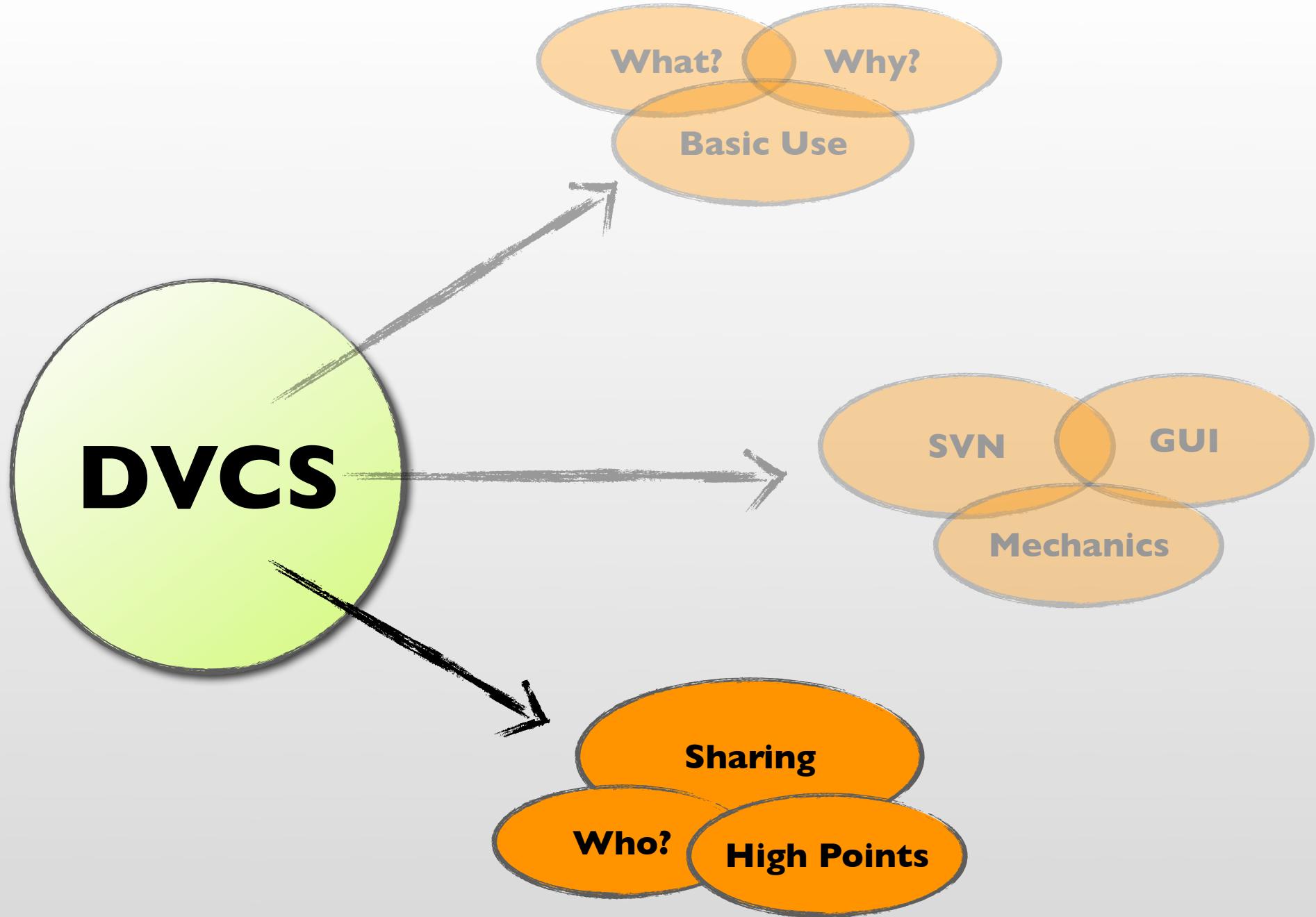
git svn clone <http://unfurl.com/trunk>

Migration

You now have a **full git clone** of the SVN repo

Migration

Power off your Subversion server



BRANCH, MERGE & REBASE

Branch

CHEAP BRANCHES

git branch newbranch

CHEAP BRANCHES

```
git checkout -b newer remote/branchnm
```

- ▶ Local branch
 - ▶ **localbranchname**
- ▶ Remote branch
 - ▶ **remote/branchname**

CHEAP BRANCHES

```
git branch -a
```

CHEAP BRANCHES

```
git show-branch --all
```

Remote & Local Branches

Remotes are **aliases** for clone URLs

Clone creates remote name '**origin**'

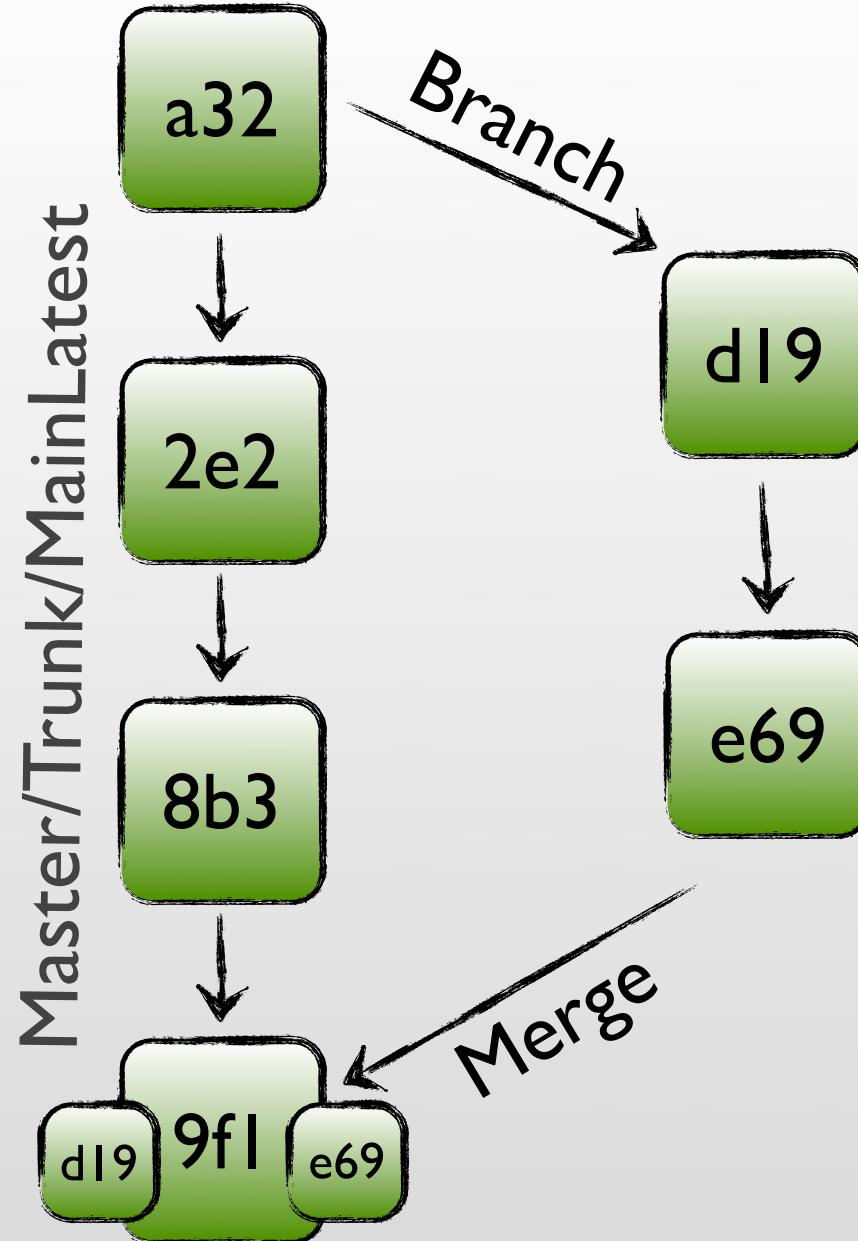
Remotes are **read-only**

But you can **track** remotes
with a **local branch**

Merge

Merges lump branch changes into a
new unified commit

Standard MERGE



MERGE Anything

```
git merge --no-commit <sourcebranch>
```

MERGE ANYTHING

```
git merge --squash <sourcebranch>
```

MERGE Anything

```
git merge <sourcebranch1> <sourcebranch2>
```

Rebase

Not a merge

Merges are old changes **woven into a new commit**

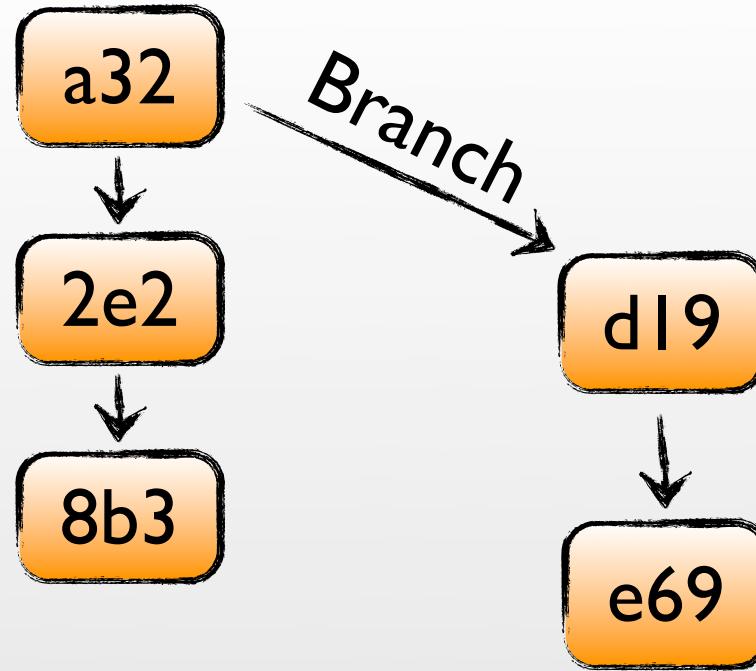
Rebase **reorders** the mainline commits **before** your branch work

REBASE for Clarity

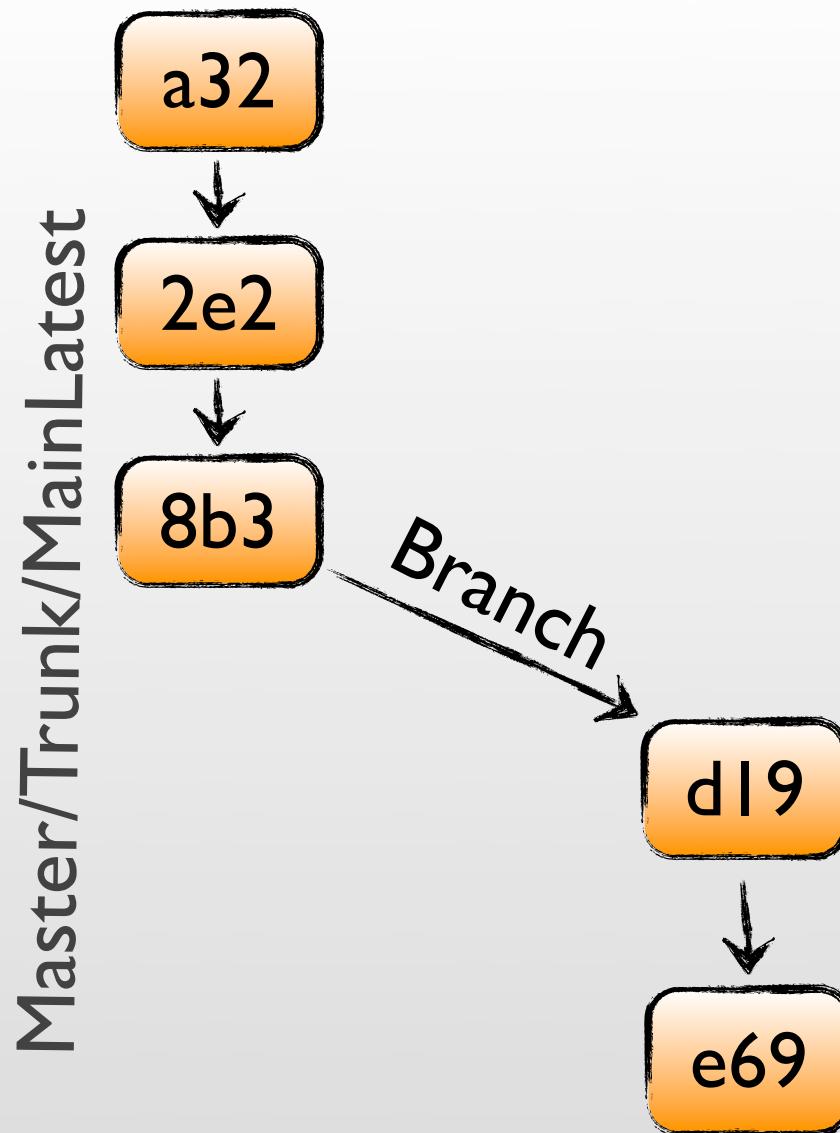
```
git checkout myfeaturebranch  
git rebase master
```

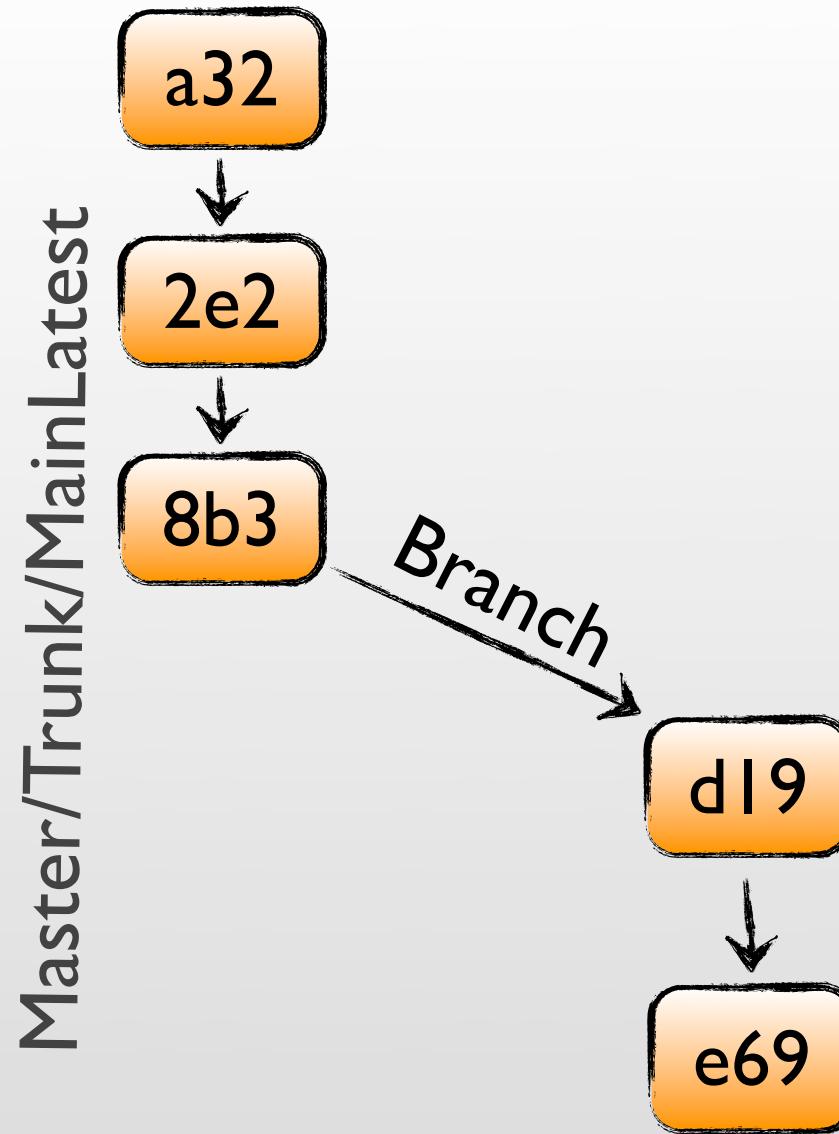
REBASING

Master/Trunk/MainLatest

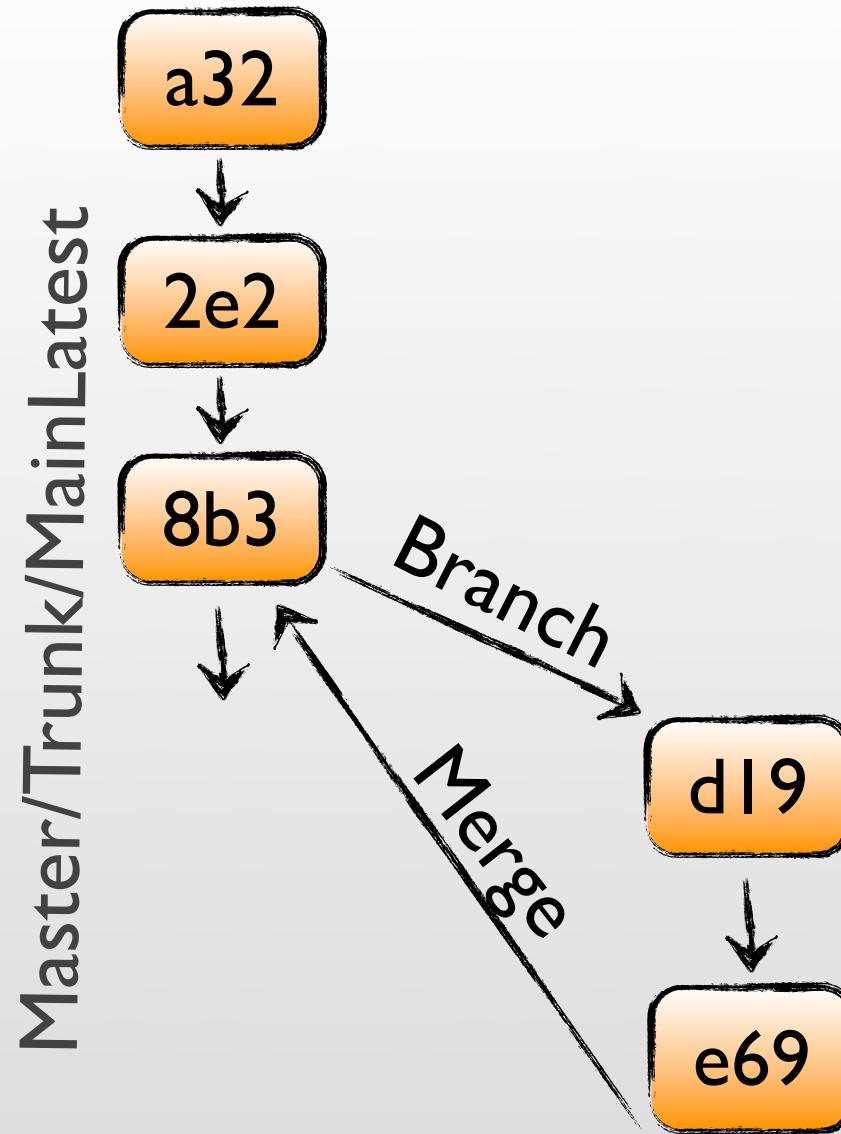


Rebase





Master/Trunk/MainLatest



Who's Got Git?



open source project

Work with the code

- [Get source](#)
- [Report bugs](#)
- [Contribute](#)
- [Code reviews](#)
- [Patch history](#)
- [Known issues](#)
- [Mailing lists](#)
- [Documentation](#)

About the project

- [Project layout](#)
- [Release features](#)
- [Roadmap](#)
- [Project roles](#)
- [Licenses](#)

Develop applications

If you are interested in developing applications for Android, visit the [Android application developer site](#).

About Android

For general information, visit [android.com](#).

Get source

For information about current problems and fixes, see [Known issues](#).

This document describes how to set up your local work environment, how to use Repo to get the Android files, and how to build the files on your machine.

Related reading:

- For an overview of the entire code-review and code-update process, see [Workflow](#).
- For reference details about Repo, see [Using Repo and Git](#).

What's in the source?

For a description of all the projects that make up the Android source code, see [Project layout](#). To see snapshots and histories of the files available in the public Android repositories, visit the [GitWeb](#) web interface.

The source is approximately 2.1GB in size. You will need 6GB free to complete the build.

Setting up your machine

To build the Android source files, you will need to use Linux or Mac OS. Building under Windows is not currently supported.

Linux



Search:

search is working now...

[Job Board](#)

[Website](#)

[Development](#)

[Enterprises is looking for a Rails Architect.](#) See more on the [Job Board](#).

Worth attention

[Workshops](#)

[Podcast](#)

Categories

Rails is moving from SVN to Git

Posted by David April 02, 2008 @ 09:39 PM

We've been preparing for Rails to move the official source repository from Subversion to [Git](#) for some time now and it seems that it'll happen over the next week or so. The premiere will happen alongside the official launch of [Github](#).

The move will also mean that we're going to be switching ticket tracking to [Lighthouse](#). So now both our repository and ticket tracking will be powered by Rails applications, which is a nice bonus treat.

When the move happens, we'll freeze the existing Subversion repository and the Trac installation. Both will live on for a long time to come, but will be entirely deprecated. This means that your existing svn:externals will not break, but if you want the latest edge, you'll have to get it from the new git repository.

So now is a great time to learn more about Git in anticipation of this move. I recommend starting with the [Git for SVN'ers crash course](#).

Perl migrates to git versioning control system.

Written on 01/03/09 at 19:14:32 by On

The [url=http://www.perlfoundation.org/]Perl Foundation[/url] has announced at the <http://use.perl.org...> website switching the version control systems to git. According to the announcement Perl 5 [url=http://use.perl.org/articles/08/12/22/0830205.shtml]migration to git[/url] would allow the language development team to take advantage of git's extensive offline and distributed versioning support.



[url=http://git.or.cz/]Git is open source[/url] and readily available to all Perl developers. Among other advantages, the announcement notes that git simplifies commits, producing fewer administrative overheads for integrating contributions. Git's change analysis tools are also singled out for praise.

The transformation from [url=http://www.perforce.com/]Perforce[/url] to git apparently took over a year. Sam Vilain of Catalyst IT "spent more than a year building custom tools to transform 21 years of Perl history into the first ever unified repository of every single change to Perl." The git repository incorporates historic snapshot releases and patch sets, which is frankly both cool and historically pleasing. Some of the patch sets were apparently recovered from old hard drives, notching up the geek satisfaction factor even more.

Git at Apache

Foundation Projects People Get Involved Support Apache Download

Git at Apache



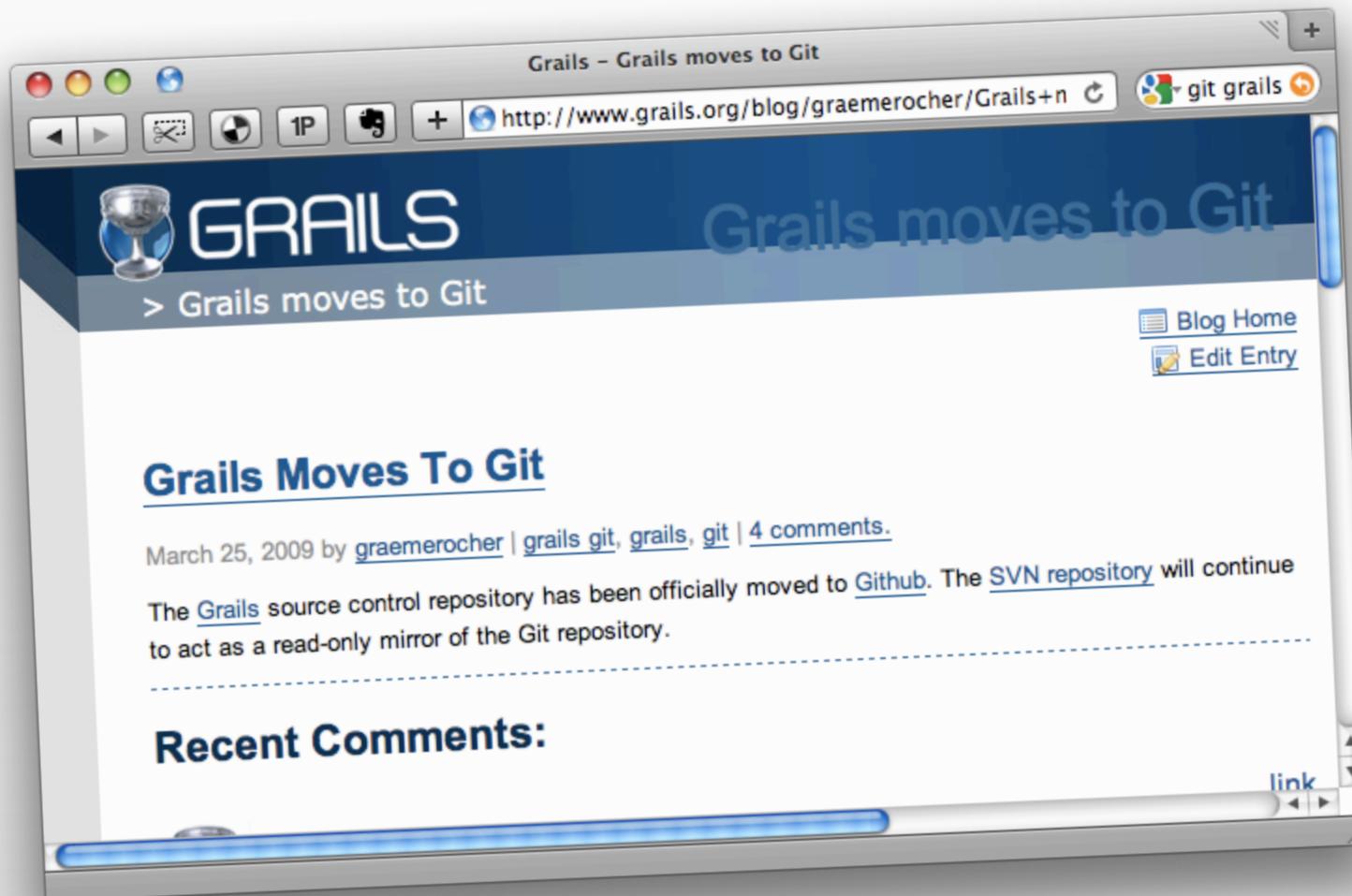
This is an collection of read-only Git mirrors of Apache codebases. The mirrors are automatically updated and contain full version histories (including branches and tags) from the respective source trees in the official Subversion repository at Apache.

Note that these Git mirrors are missing Subversion features like svn:ignore, svn:eol-style and svn:keywords settings and support for empty directories.

Please contact the infrastructure-dev@apache.org mailing list if you have comments or suggestions regarding this service or want to see another Apache codebase mirrored.

Project	Description	Git Clone URL	HTTP Clone URL
abdera.git	Apache Abdera	git://git.apache.org/abdera.git	http://git.apache.org/abdera.git
activemq.git	Apache ActiveMQ	git://git.apache.org/activemq.git	http://git.apache.org/activemq.git
ant.git	Apache Ant	git://git.apache.org/ant.git	http://git.apache.org/ant.git
apr-iconv.git	Apache Portable Runtime iconv	git://git.apache.org/apr-iconv.git	http://git.apache.org/apr-iconv.git
apr-util.git	Apache Portable Runtime util	git://git.apache.org/apr-util.git	http://git.apache.org/apr-util.git
apr.git	Apache Portable Runtime	git://git.apache.org/apr.git	http://git.apache.org/apr.git
archiva.git	Apache Archiva	git://git.apache.org/archiva.git	http://git.apache.org/archiva.git
axis2-c.git	Apache Axis2/C	git://git.apache.org/axis2-c.git	http://git.apache.org/axis2-c.git
axis2-java.git	Apache Axis2/Java	git://git.apache.org/axis2-java.git	http://git.apache.org/axis2-java.git
batik.git	Apache Batik	git://git.apache.org/batik.git	http://git.apache.org/batik.git
buildr.git	Apache Buildr	git://git.apache.org/buildr.git	http://git.apache.org/buildr.git
camel.git	Apache Camel	git://git.apache.org/camel.git	http://git.apache.org/camel.git
cassandra.git	Apache Cassandra (incubating)	git://git.apache.org/cassandra.git	http://git.apache.org/cassandra.git
cayenne.git	Apache Cayenne	git://git.apache.org/cayenne.git	http://git.apache.org/cayenne.git
chainsaw.git	Apache Chainsaw	git://git.apache.org/chainsaw.git	http://git.apache.org/chainsaw.git
cocoon.git	Apache Cocoon	git://git.apache.org/cocoon.git	http://git.apache.org/cocoon.git
commons-codec.git	Apache Commons Codec	git://git.apache.org/commons-codec.git	http://git.apache.org/commons-codec.git
commons-collections.git	Apache Commons Collections	git://git.apache.org/commons-collections.git	http://git.apache.org/commons-collections.git

Go to "<http://git.apache.org/maven.git>"



CULTURE CHANGE



integration builds

CENTRALIZED
VCS = DON'T

DISTRIBUTED
VCS = Do!

Safely Be Wrong



Experiment



Innovate



Drive-by Assist



Crowd Source



3
reasons

CHERRY Pick

- ▶ `git cherry-pick a5b2ee`
- ▶ Merge in just one commit.

SEARCH History

- ▶ `git grep SomeText HEAD^{~3}`
- ▶ Search blob contents history without checkouts.

Bisect Bugs

- ▶ `git bisect run mvn test`
- ▶ Binary-search for bug via failed tests.
- ▶ Manual or automated modes.
- ▶ Ant, Maven, or shell script that can return 0 or 1-127

*Feedback is greatly
appreciated*

Twitter
@matthewmccull

Blog
<http://www.ambientideas.com/blog>
sidebar has all my social media links

Email
matthewm@ambientideas.com

GitHub
<http://github.com/matthewmccullough>



Discussion IDEAS

Partial Commits

- ▶ Selecting a piece of the file changes to commit

Interactive Commit

- ▶ Selecting a piece of the file changes to commit
- ▶ git add -i
- ▶ git add -p

Discussion IDEAS

Gitosis

- ▶ Securing repos by unique SSH keys, under only one system user account.

Git namespaces

- ▶ Namespace for branches is separate for each repo
- ▶ Unlike Mercurial which shares the namespace across all nodes

Git interactive add of files, full menu prompt

- ▶ `git add --interactive`

Git interactive add of files, patch menu options

- ▶ `git add -p`

Discussion IDEAS

Tracking branches vs local branches

- ▶ Cat the branches file to show the tracking
- ▶ cat .git/config

Clone a repo to a partial depth

- ▶ git clone --depth 2 <repo-addr>
- ▶ git-clone manpage can be found at <http://git.or.cz/man/git-clone>
- ▶ You won't be able to push/pull, although I think you can still commit.

Gerrit

- ▶ <http://code.google.com/p/gerrit/>
- ▶ Web based code review and project management for Git-based projects.
- ▶ Gerrit makes reviews easier by showing changes in a side-by-side display
- ▶ Allows inline comments to be added by any reviewer.
- ▶ Simplifies project maintainership by permitting any authorized user to submit changes to the master Git repository, rather than requiring all approved changes to be merged in by hand by the project maintainer.
- ▶ Enables a more centralized usage of Git.
- ▶ Example review:
 - ▶ <https://review.source.android.com/Gerrit#all,open,n,z>

RESOURCES

Git Homepage

<http://git-scm.com>

Directed Acyclic Graph

http://en.wikipedia.org/wiki/Directed_acyclic_graph

Git for Computer Scientists

<http://eagain.net/articles/git-for-computer-scientists/>

Git Treeish

http://book.git-scm.com/4_git_treeishes.html

RESOURCES

Git Docs

<http://www.kernel.org/pub/software/scm/git/docs/>

Git Magic eBook

<http://www-cs-students.stanford.edu/~blynn/gitmagic/book.pdf>

Linus Torvald's Git talk at Google

<http://www.youtube.com/watch?v=4XpnKHJAok8>

CygWin

<http://www.cygwin.com>

RESOURCES

MSysGit

<http://code.google.com/p/msysgit>

Git Cheetah

<https://git.wiki.kernel.org/index.php/MSysGit:GitCheetah>

Matthew's Git Bookmarks

<http://delicious.com/matthew.mccullough/git>

Matthew's Bash Prompt

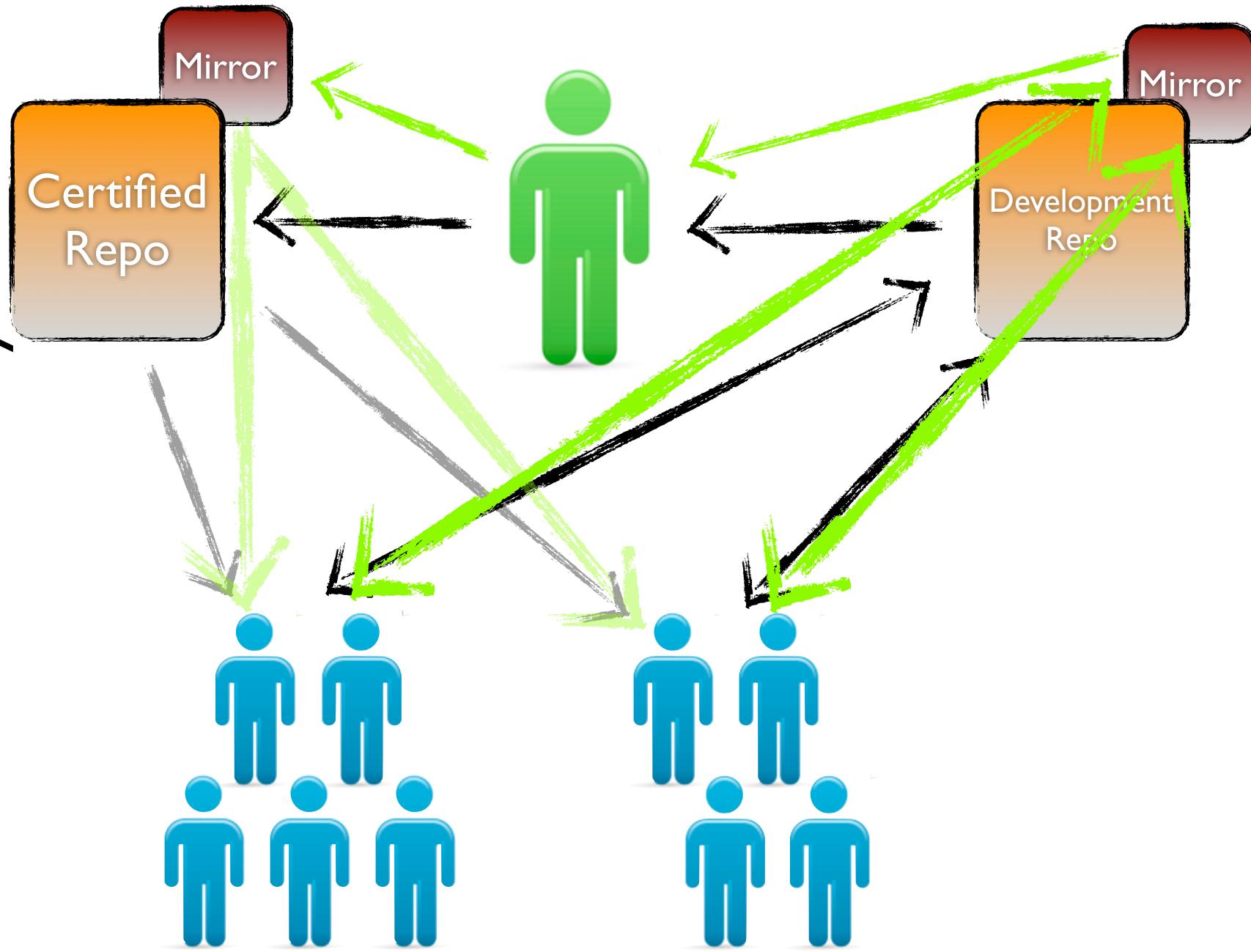
<http://gist.github.com/47267>

IMAGE CREDITS

- ▶ <http://www.ambientideaspotography.com>
- ▶ <http://flickr.com/photos/lenore-m/2903856664/>
- ▶ [http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software))
- ▶ <http://flickr.com/photos/karenhorton/1583513014/>
- ▶ <http://flickr.com/photos/mashdnart/2545782407/>
- ▶ [http://commons.wikimedia.org/wiki/
File:Small_Boy_nuclear_test_1962.jpg](http://commons.wikimedia.org/wiki/File:Small_Boy_nuclear_test_1962.jpg)
- ▶ <http://www.flickr.com/photos/knmurphy/2506896257/>
- ▶ <http://www.flickr.com/photos/albyspace/1022035568/>
- ▶ <http://flickr.com/photos/michaelhays/3070238360/>
- ▶ http://flickr.com/photos/d_vdm/509996632/

BONUS MATERIALS

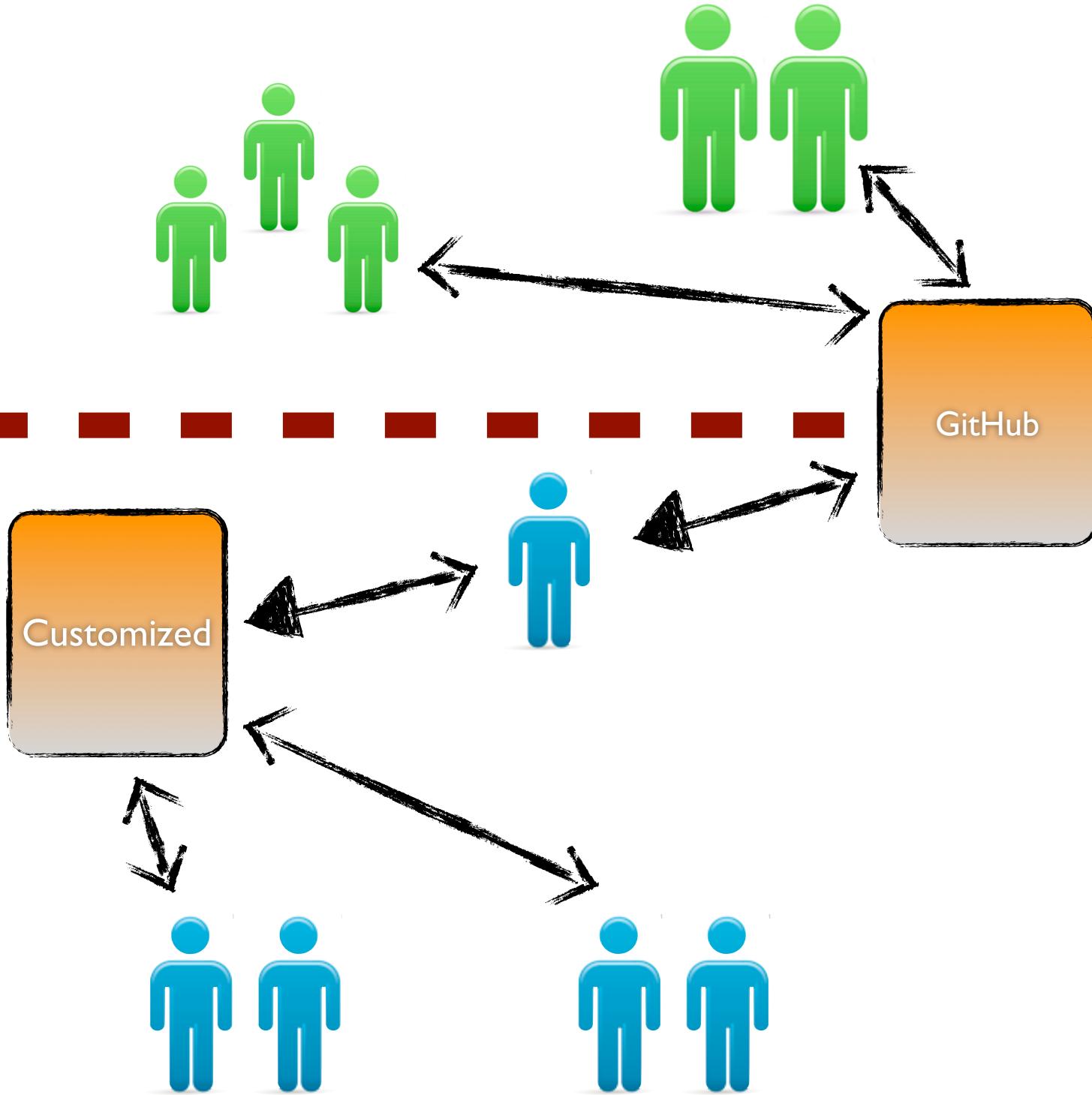
Bonus Self-Study Materials



Mirrored

Custom + Public Contribution

Public ↗
Private ↘



Git EXPORTING by FILE

```
git bundle create mydiff.bundle 82AB2..HEAD
```

Hash Collisions

Bonus Self-Study Materials

But I've been
hearing news
about hash
collisions...

10^{49}

earth atoms

Bonus Self-Study Materials

10^{46}

SyAI

Bonus Self-Study Materials

10^{22}
stars

Bonus Self-Study Materials

10^{20}

sand grains

10^{-9}
Collision

SHA-1 Hash

- ▶ Unlike hashing a password.
- ▶ Keep the source of the hash.

WHAT do REVERTABLE Commits
buy ME?

Bonus Self-Study Materials

```
$ echo //MYD0GSNAM3 >> mattfile1.txt
$ echo //NewText2 >> mattfile2.txt

$ git status
# On branch master
# Changed but not updated:
#   modified:   mattfile1.txt
#   modified:   mattfile2.txt

$ git add mattfile1.txt Put into index
$ git commit -m'Matts pw checkin' Save to repo
[master]: created ddcdf18: "Matts pw checkin"
1 files changed, 1 insertions(+), 0 deletions(-)
```

#\$%&!

I just committed my password!

CHANGE History

```
$ git reset --hard HEAD^  
HEAD is now at 9de4cd3 First checkin
```

```
$ git log  
commit 9de4cd3a3f24a3de3e804df84cc3321c7d0c9993  
Author: Matthew McCullough  
<matthewm@ambientideas.com>  
Date:   Wed Mar 4 23:09:17 2009 -0700  
First checkin
```

Throw away the commit

Bonus Self-Study Materials

SHARING Tools



SERVING IT UP

Protocols

- ▶ **GIT**
- ▶ **SSH**
- ▶ SCP
- ▶ Local file system
- ▶ SAMBA
- ▶ HTTP
- ▶ WebDAV
- ▶ rsync

Bonus Self-Study Materials

- git **clone** git://somehost.org/**myproj.git**  *git daemon*
- git **clone** mccm06@somehost.org:**myproj.git**  *SSH*
- git **clone** http://somehost.org/**myproj.git**  *HTTP*
- git **clone** ~/work/**myproj.git**  *Filesystem*
- git **clone** z:\someserver\myproj.git  *Network drive*

SERVING IT UP

- ▶ **git daemon.** Custom socket language.
- ▶ **gitosis.** Easy SSH-based repo permissions.
- ▶ **ssh.** OS controlled permissions.

SERVING IT UP

- ▶ **git instaweb**. Built in, read-only, HTTP.

Bonus Self-Study Materials

localhost Git - .git/summary

[RSS](#) [Atom](#) Google

[projects](#) / [.git](#) / [summary](#)

[summary](#) | [shortlog](#) | [log](#) | [commit](#) | [commitdiff](#) | [tree](#)

commit search: re

description Unnamed repository; edit this file to name it for gitweb.
owner Matthew McCullough
last change Sat, 28 Feb 2009 21:23:06 +0000

shortlog

Time Ago	Author	Commit Message	Branches	Links
111 min ago	Matthew McCullough	Bob file added text	master origin/HEAD origin/master	commit commitdiff tree snapshot
117 min ago	Matthew McCullough	Merge branch 'aliceonbob'		commit commitdiff tree snapshot
117 min ago	Matthew McCullough	Merge branch 'bobs2ndidea' into aliceonbob	aliceonbob origin/aliceidea origin/aliceonbob	commit commitdiff tree snapshot
118 min ago	Matthew McCullough	Alice workong on bobs branch		commit commitdiff tree snapshot
119 min ago	Matthew McCullough	Masterwork three.		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Bobs second idea	bobthesecond	commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Masterwork again		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Bob masterwork		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Bobs idea		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Merge branch 'aliceidea'		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Alice masterwork		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Alices idea		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	Bobs checkin		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	First checkin for alice		commit commitdiff tree snapshot
2 hours ago	Matthew McCullough	First checkin;		commit commitdiff tree snapshot

heads

Time Ago	Branch	Links
111 min ago	master	shortlog log tree
117 min ago	aliceonbob	shortlog log tree
2 hours ago	bobthesecond	shortlog log tree

Unnamed repository; edit this file to name it for gitweb.

Display a menu

SERVING IT UP

- ▶ **github.** Open source & private repos plus innovative visualizations.

Bonus Self-Study Materials

matthewmccullough's hellogitworld at master - GitHub

git <http://github.com/matthewmccullough/hellogitworld/tree/master>

RSS Google

github SOCIAL CODING Search

Browse | Guides | Advanced

matthewmccullough account | profile | log out
dashboard | gists

Source Commits Network (1) Fork Queue Downloads (0) Wiki (1) Graphs Admin

master all branches all tags

matthewmccullough / hellogitworld edit pull request unwatch download

Description: Hello World for Presentation Demos edit

Homepage: <http://www.ambientideas.com> edit

Public Clone URL: <git://github.com/matthewmccullough/hellogitworld.git>

Your Clone URL: <git@github.com:matthewmccullough/hellogitworld.git>

Added a new file

matthewmccullough (author) January 14, 2009

commit 4e0d383949974d9bd6ff977980bb991794b717ba
tree 1b8cf0e780b227472cb8c89697e7f4429f34d531
parent 0bbe90b062b4a75f7a0c87903a232c9a058b2f2b

hellogitworld /

name	age	message	history
anewfile.txt	January 14, 2009	Added a new file [matthewmccullough]	
another.txt	December 14, 2008	A fix by Matthew [matthewmccullough]	
text.txt	January 14, 2009	Changes to text [matthewmccullough]	

[Shirts](#) | [Blog](#) | [Support](#) | [Training](#) | [Contact](#) | [Google Group](#) | [API](#) | [Status](#)

GitHub is Logical Awesome ©2009 | [Terms of Service](#) | [Privacy Policy](#)

Hosting provided by our partners at Engine Yard

Display a menu

SERVING IT UP

Filesystem

- ▶ Local folders. Yes, they can be “remotes.”
- ▶ Mapped network drives. Just like folders.

Bash Prompt

- Shell that tells me < or > or changes and branch name
- <http://gist.github.com/48058>
Windows
- <http://gist.github.com/47267>
Unix

Global Git Settings

git config --list

Output current values

REBASE FOR CLARITY

Bonus Self-Study Materials

- ▶ Fast forwards the “side” branch.
- ▶ Reposition your feature branch’s start point.
- ▶ See “how trunk will act” before you merge.
- ▶ Makes for simpler merge graphs.
- ▶ Cleaner history than a merge.

MERGE Anything

Bonus Self-Study Materials

- ▶ Selectable merge strategy
- ▶ Octopus: More than one

MERGE STRATEGIES

MERGE STRATEGIES

resolve

This can only resolve two heads (i.e. the current branch and another branch you pulled from) using 3-way merge algorithm. It tries to carefully detect criss-cross merge ambiguities and is considered generally safe and fast.

recursive

This can only resolve two heads using 3-way merge algorithm. When there are more than one common ancestors that can be used for 3-way merge, it creates a merged tree of the common ancestors and uses that as the reference tree for the 3-way merge. This has been reported to result in fewer merge conflicts without causing mis-merges by tests done on actual merge commits taken from Linux 2.6 kernel development history. Additionally this can detect and handle merges involving renames. This is the default merge strategy when pulling or merging one branch.

octopus

This resolves more than two-head case, but refuses to do complex merge that needs manual resolution. It is primarily meant to be used for bundling topic branch heads together. This is the default merge strategy when pulling or merging more than one branches.

ours

This resolves any number of heads, but the result of the merge is always the current branch head. It is meant to be used to supersede old development history of side branches.

subtree

This is a modified recursive strategy. When merging trees A and B, if B corresponds to a subtree of A, B is first adjusted to match the tree structure of A, instead of reading the trees at the same level. This adjustment is also done to the common ancestor tree.

If you tried a merge which resulted in a complex conflicts and would want to start over, you can recover with *git-reset*.

Interactive versions of rebase

SQUASH VERBOSITY

- ▶ `git merge --squash mybranch`
- ▶ Reduce all your micro-checkins to block commits.
- ▶ Make mistakes, revise, checkin without hesitation.
- ▶ Great for SVN compatibility.

Bonus Self-Study Materials

The screenshot shows a web browser window with the title "Why Git is Better Than X". The URL in the address bar is <http://whygitisbetterthanx.com/>. The page content lists several reasons why Git is better than other version control systems (X), each with a "More" link and a "Details" link. The reasons are:

- Cheap Local Branching
- Everything is Local
- Git is Fast
- Git is Small
- The Staging Area
- Distributed
- Any Workflow
- GitHub

Each reason has a set of colored buttons at the bottom right: hg (blue), bzr (yellow), svn (purple), and perforce (red). A note above the "GitHub" section states "where 'X' is one of hg, bzr, svn, or perforce".

Gives Credit

Bonus Self-Study Materials

- ▶ Tracks who authors a change.
- ▶ “Author” even survives passing through reviewers.
- ▶ “Author” survives all merges.
- ▶ Additionally tracks who committed a change.