



# Git Introduction

For Subversion Users  
Ran Tavory  
Attributed to Totango

# Who am I



- A Developer, of course...
- Migrated two teams from SVN to Git
- Gormim



# We help developers find their dream job!

Curated weekly email of top companies seeking talented software engineers in Israel

Subscribe



Nir Dremer

Nir builds products at the intersection of technology, business & psychology. He founded MediaBox and led products at Imperva (NYSE: IMPV) & BitBand

(acquired by Motorola)



Ran Tavory

Ran is domain expert in web infrastructure and scaling. He founded SocialStudios and built products at Google, YouTube, Microsoft and Outbrain. Ran is

also a blogger and a podcaster.



**Why git?**

# Why Git?

- IT IS BETTER. There's a site for that.

## Why Git is Better than X

hg bsr svn perforce

where "X" is one of ↻

This site is here because I seem to be spending a lot of time lately defending Gitsters against charges of fanboyism, bandwagonism and koolaid-thirst. So, here is why people are switching to Git from X, and why you should too. Just click on a reason to view it.

[Expand all](#) | [Collapse all](#)

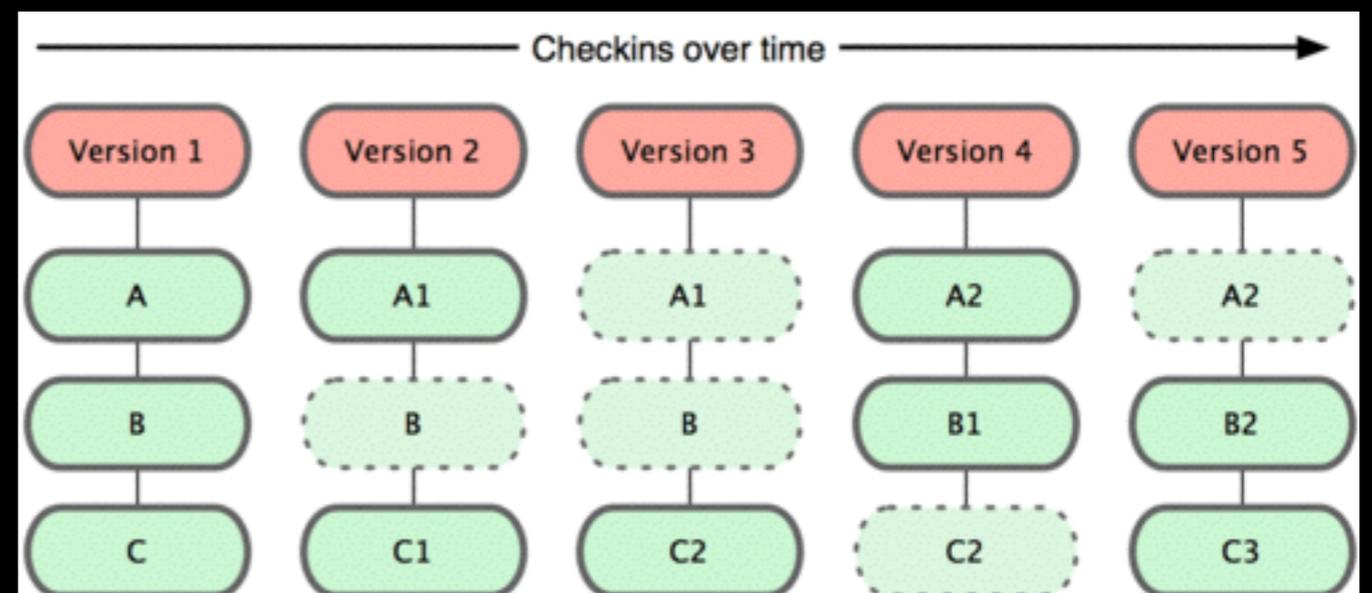
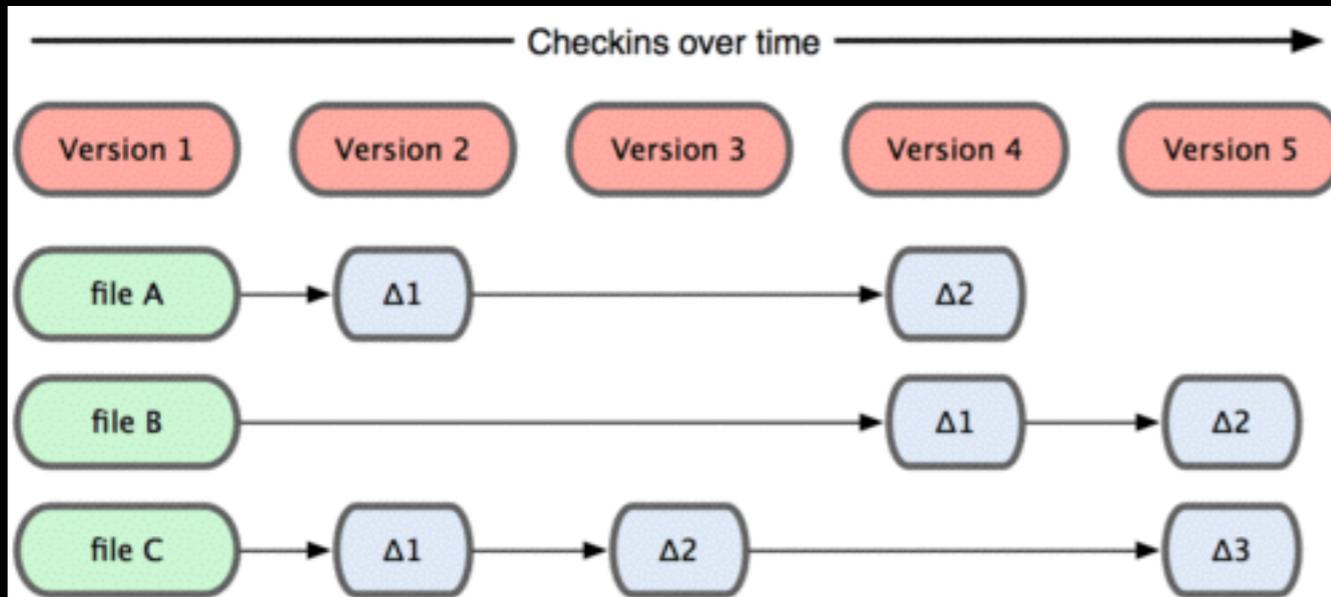
Cheap Local Branching

hg bsr svn perforce

Everything is Local

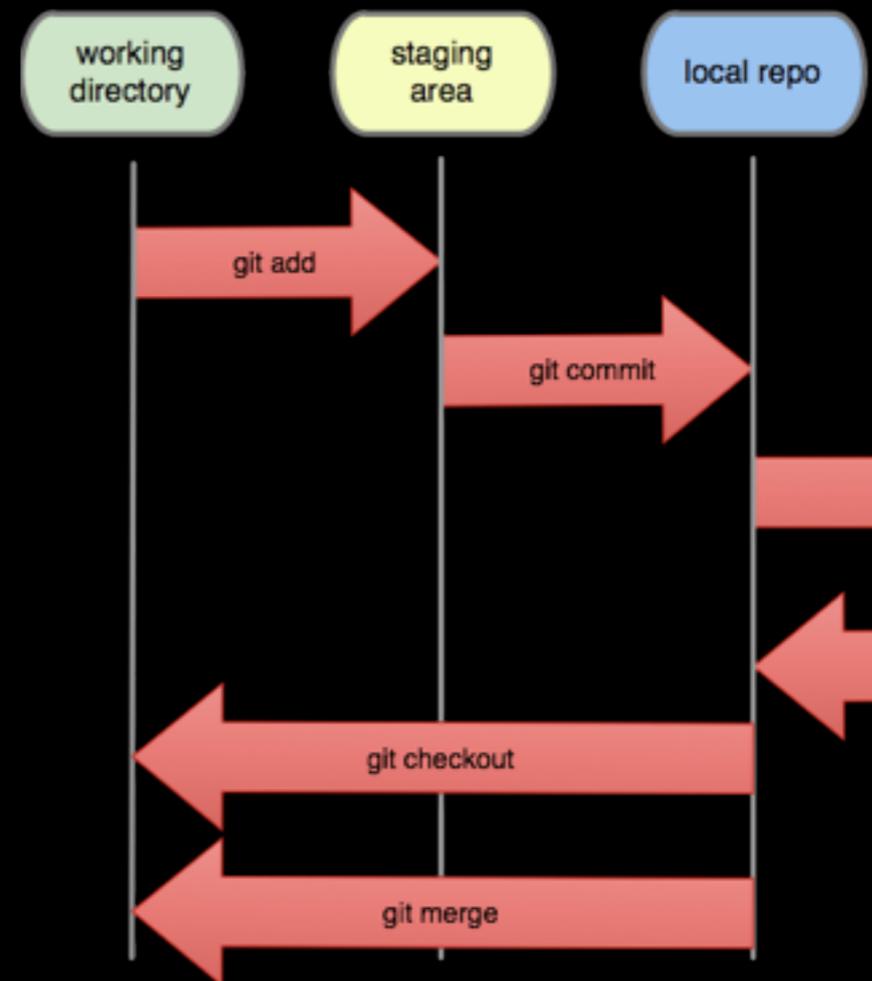
svn perforce

# Git is a File System

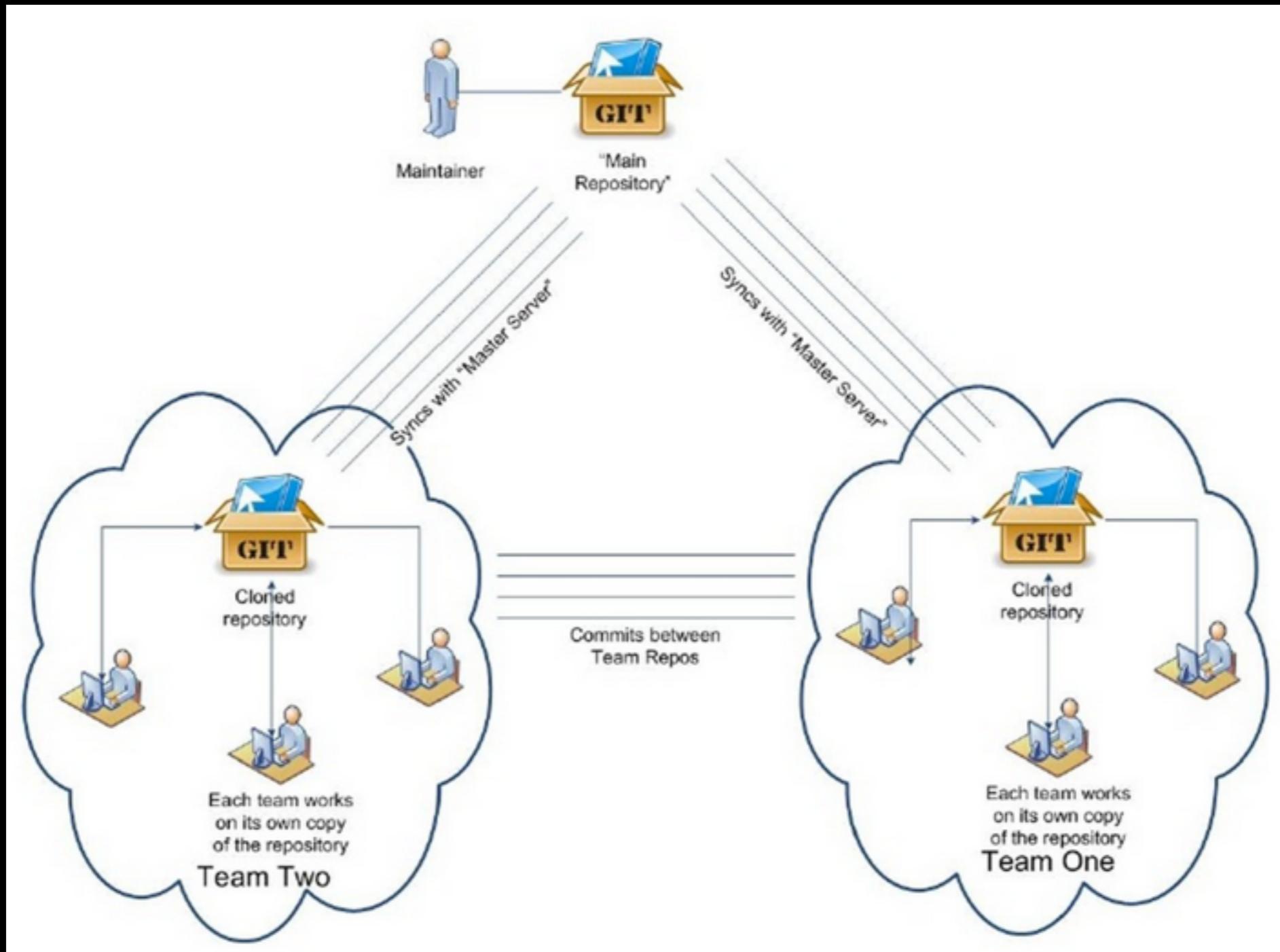


# All (most) Operations are Local and Fast!

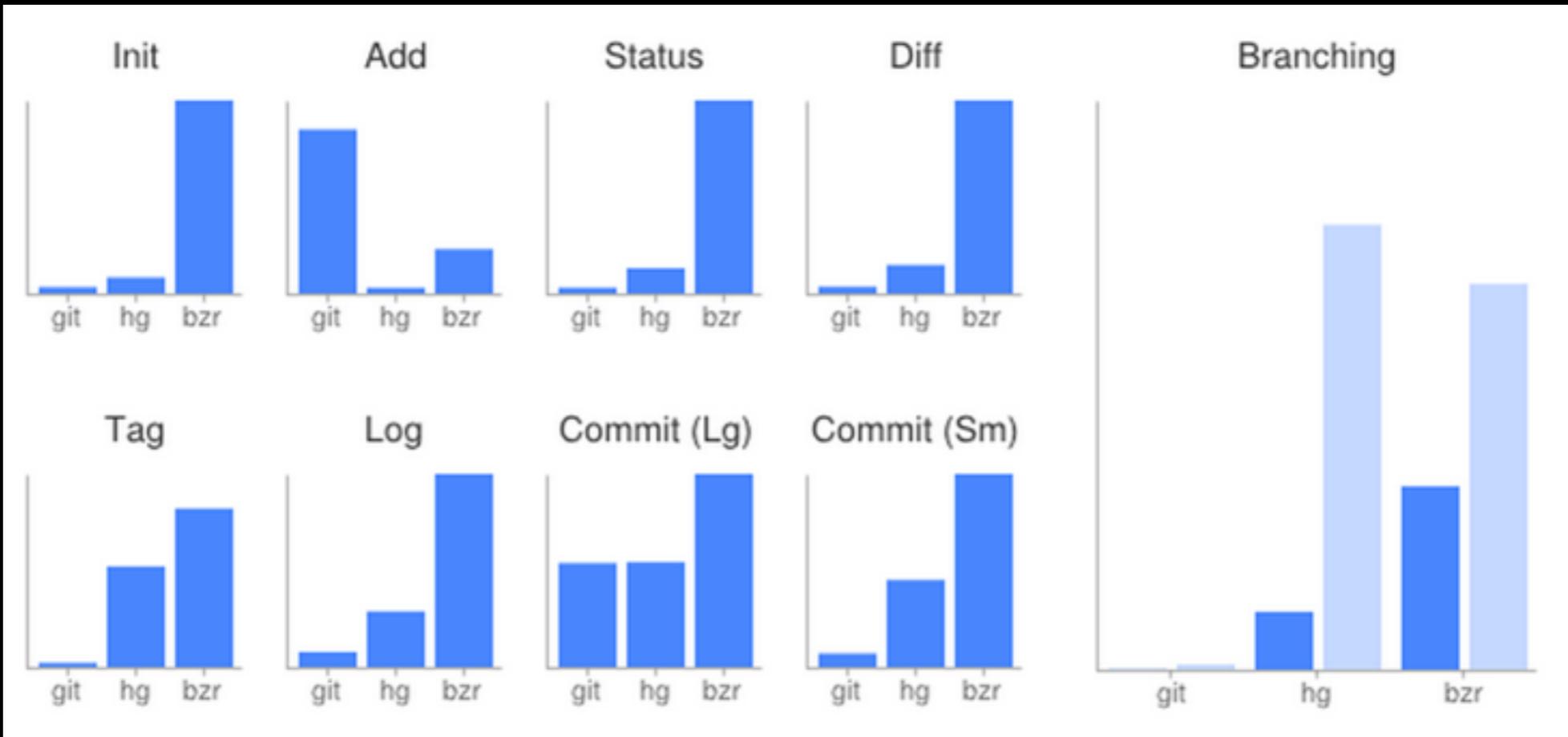
- Commit
- Log (view history)
- Diff
- ...



# Git is Distributed



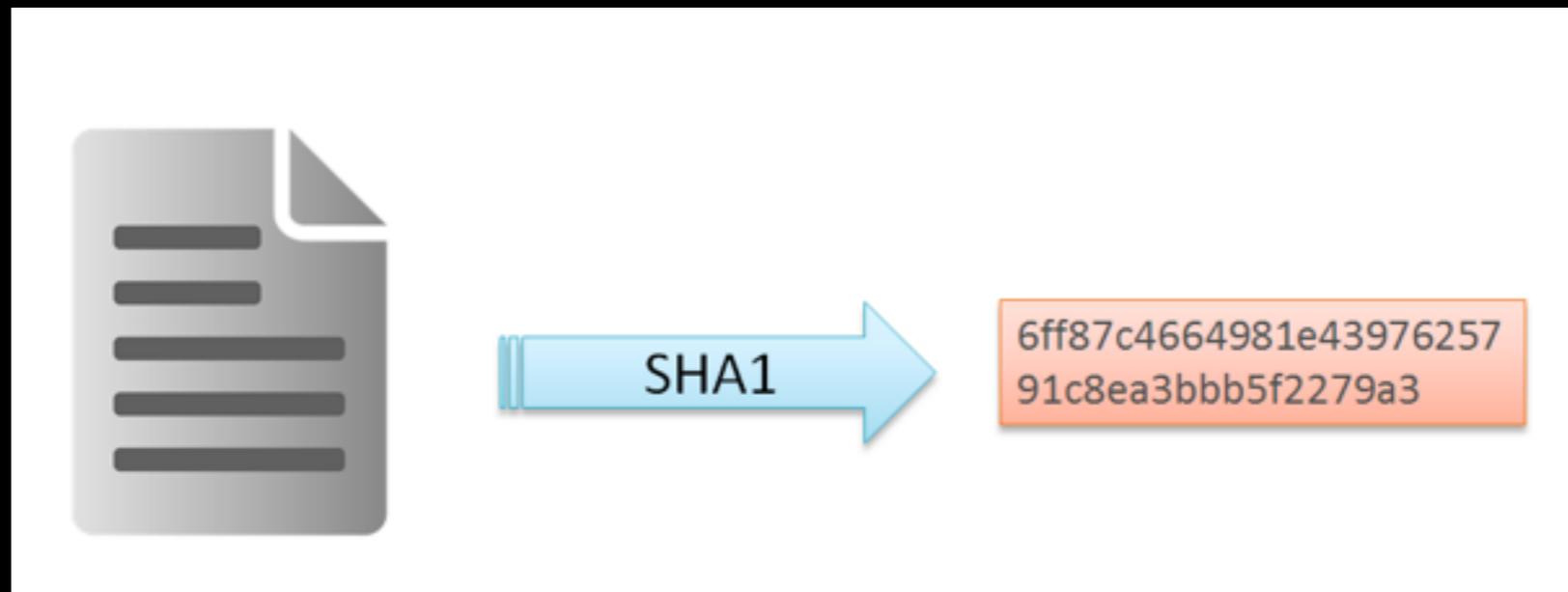
# Git is Fast



# Lightweight (and local) Branching

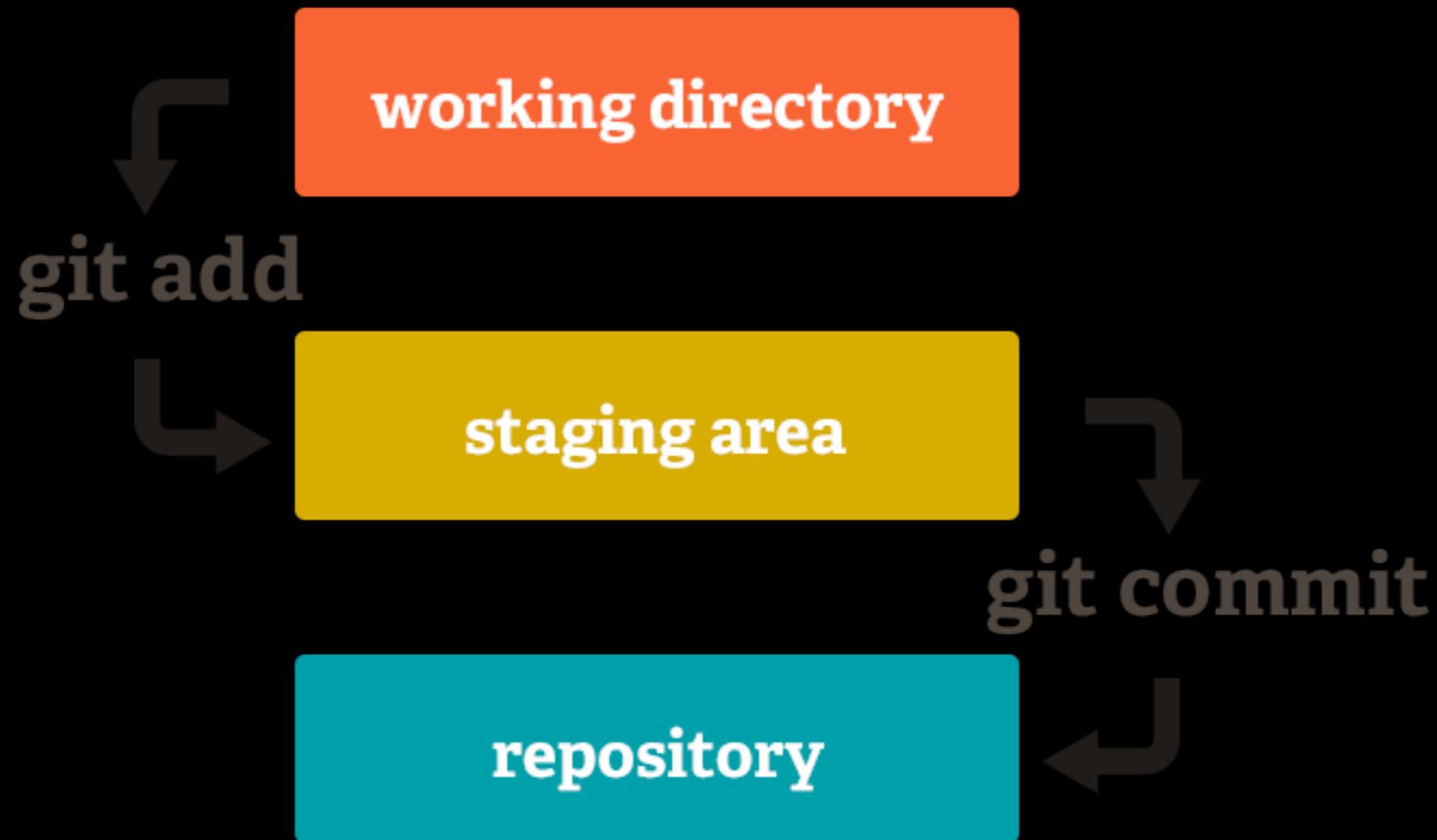


# Data Integrity



```
(fabric)~/dev/totango/main/gameboy (master*) $ git rebase
4d93f85 HEAD@{0}: commit: Add support for fb-flo for hot relaod. Still very epxperimental
22ade50 HEAD@{1}: commit: Improve the loading animation of the Jenkins button.
455fc15 HEAD@{2}: commit: Add to gameboy luigi_external_url to fix the links to luigi with
862eaea HEAD@{3}: reset: moving to refs/remotes/git-svn
36ae50c HEAD@{4}: commit: Remove the immune project from parent/pom.xml. It doesn't need to
c925399 HEAD@{5}: reset: moving to refs/remotes/git-svn
c1de30b HEAD@{6}: commit: Update newrelic's version to 3.5.1
302eef2 HEAD@{7}: rebase finished: returning to refs/heads/master
```

# Staging Area



# Stash Area





An aerial, black and white photograph of a large crowd of people at what appears to be a festival or concert. The people are seen from above, some standing and some sitting on the ground. The word "Setup" is overlaid in a large, white, sans-serif font in the center of the image. The background is dark and filled with the silhouettes of many people, creating a sense of a large gathering.

# Setup

# Git Setup

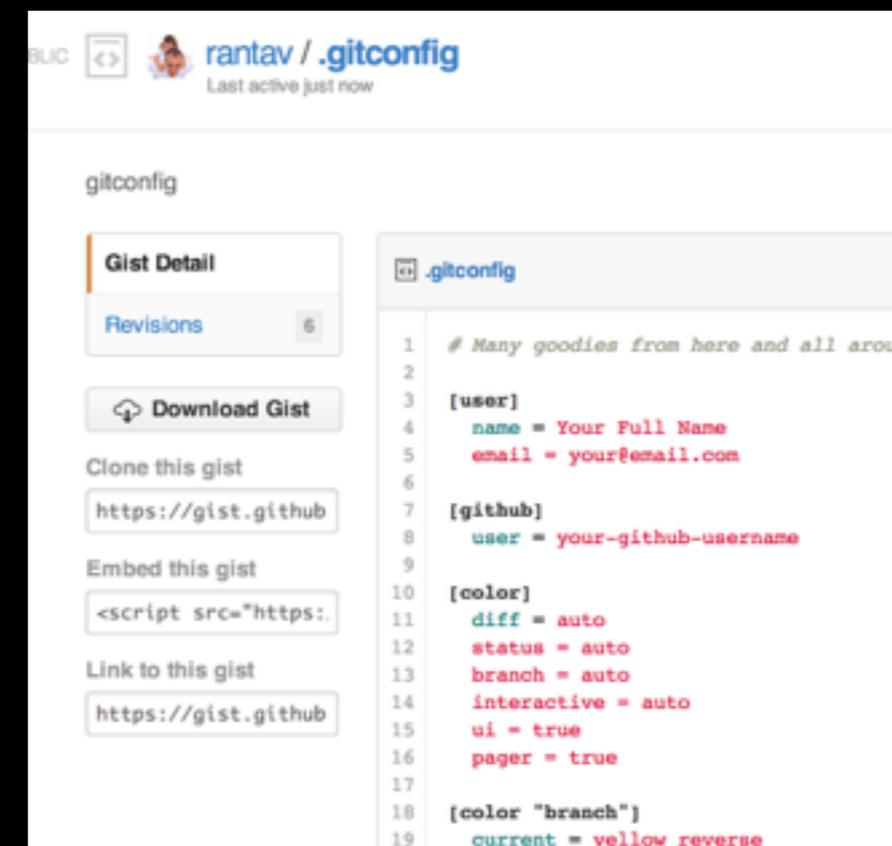
```
$ brew install git
```

```
$ git config --global user.name "Ran Tavory"
```

```
$ git config --global user.email "rantav@gmail.com"
```

<https://gist.github.com/rantav/5d6fb14057f062ecbc70>

```
$ git config --global alias.co checkout  
$ git config --global alias.br branch  
$ git config --global alias.ci commit  
$ git config --global alias.st status
```



# GitHub Setup

[https://help.github.com/articles/  
set-up-git](https://help.github.com/articles/set-up-git)

Bootcamp / Set Up Git

Article last updated on 09-May-14

## Set Up Git

MAC | WINDOWS | LINUX | ALL



Forget the terminal. Download our native

If you've found yourself on this page, we're assuming you've never used Git before. We'll walk you through the basics and explain a little bit about how it works.

### Download and Install Git

At the heart of GitHub is an open source version control system. The same team that created Linux, Git is responsible for the software that runs your computer.

*\*If you don't already know what Git is, [take a crash course](#)*

Download and install [the latest version of Git](#)

# Clone the Repo

```
$ git clone https://github.com/  
rantav/devdev.git
```

**HTTPS** clone URL

https://github.com



You can clone with [HTTPS](#), [SSH](#),  
or [Subversion](#). 



# The Basics

# git status

```
$ git status  
On branch master  
nothing to commit, working directory clean
```

# git add

```
$ git add benchmarks.rb
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   README
        modified:  benchmarks.rb
```

# git diff

```
$ git diff
diff --git a/benchmarks.rb b/benchmarks.rb
index 3cb747f..da65585 100644
--- a/benchmarks.rb
+++ b/benchmarks.rb
@@ -36,6 +36,10 @@ def main
     @commit.parents[0].parents[0].parents[0]
   end

+   run_code(x, 'commits 1') do
+     git.commits.size
+   end
+
   run_code(x, 'commits 2') do
     log = git.commits('master', 15)
     log.size
```

# git diff

```
$ git diff
diff --git a/README b/README
index bccdfbd..b0ed415 100644
--- a/README
+++ b/README
@@ -1,2 @@
 This is the README file.
+One more line.
```

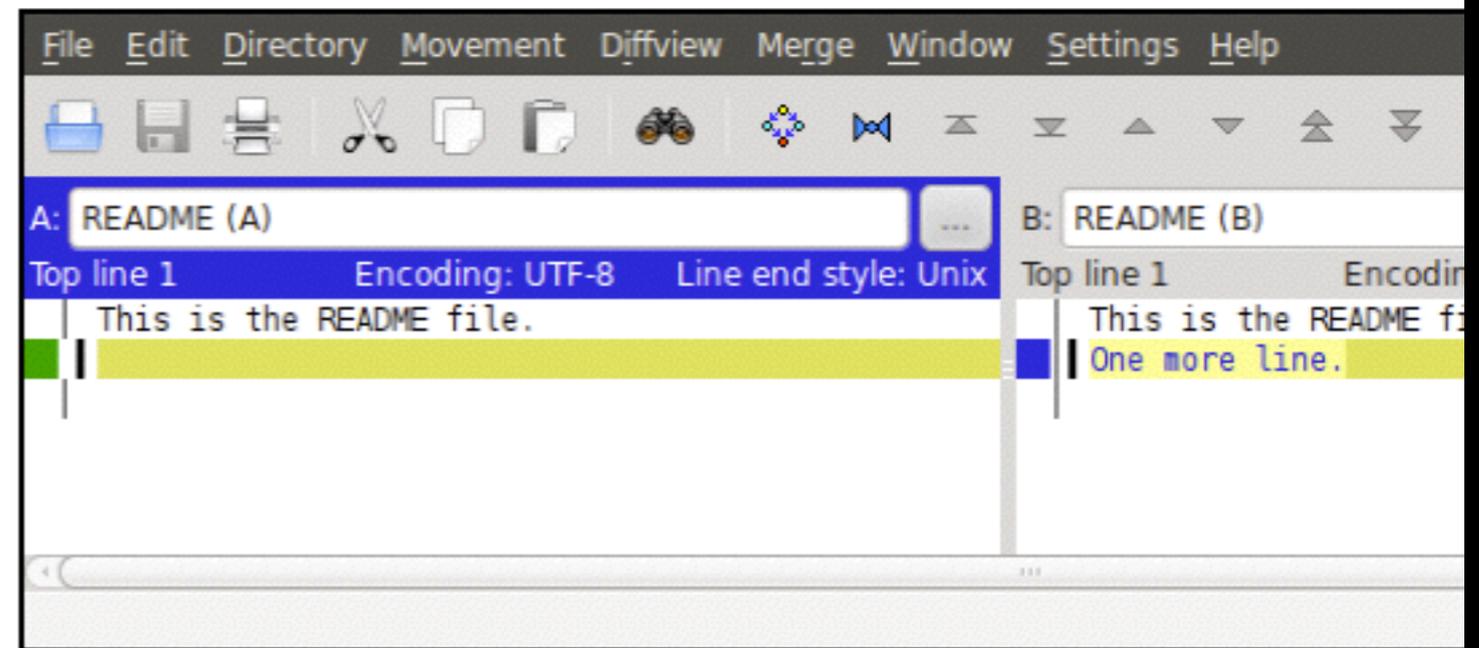
```
$ git diff --cached
```

```
$ git diff HEAD
diff --git a/README b/README
index bccdfbd..b0ed415 100644
--- a/README
+++ b/README
@@ -1,2 @@
 This is the README file.
+One more line.
```

```
$ git difftool
```

```
Viewing: 'README'
Hit return to launch 'kdiff3':
```

The following window will then be displayed showing the differences between the README file in the working directory (on the left) compared with the **README** file in the working directory (on the right):



More here: <http://www.gitguys.com/topics/git-diff/>

# git commit

```
$ git commit
```

```
# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch master
# Changes to be committed:
#   new file:   README
#   modified:   benchmarks.rb
#
~
~
~
".git/COMMIT_EDITMSG" 10L, 283C
```

```
$ git commit -m "Story 182: Fix benchmarks for speed"
[master 463dc4f] Story 182: Fix benchmarks for speed
2 files changed, 3 insertions(+)
create mode 100644 README
```

# git commit --amend

```
$ git commit -m 'initial commit'  
$ git add forgotten_file  
$ git commit --amend
```

# git checkout

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified: benchmarks.rb
```

```
$ git checkout -- benchmarks.rb
```

# git rm

```
$ rm grit.gemspec
$ git status
On branch master
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will
  (use "git checkout -- <file>..." to discard ch

        deleted:    grit.gemspec

no changes added to commit (use "git add" and/or
```

```
$ git rm grit.gemspec
```

```
$ git rm log/\*.log
```

# git mv

```
$ git mv file_from file_to
```

# git log

```
$ git log
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700
```

# git log -p

```
$ git log -p -2
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -5,5 +5,5 @@ require 'rake/gempackagetask'
  spec = Gem::Specification.new do |s|
    s.name       = "simplegit"
-   s.version    = "0.1.0"
+   s.version    = "0.1.1"
    s.author     = "Scott Chacon"
    s.email      = "schacon@gee-mail.com"
```

# git log --word-diff

```
$ git log -U1 --word-diff
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

diff --git a/Rakefile b/Rakefile
index a874b73..8f94139 100644
--- a/Rakefile
+++ b/Rakefile
@@ -7,3 +7,3 @@ spec = Gem::Specification.new do |s|
   s.name       = "simplegit"
   s.version    = ["0.1.0-"]{+"0.1.1"+}
   s.author     = "Scott Chacon"
```

# git log --stat

```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    changed the version number

Rakefile |      2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608ele8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    removed unnecessary test code

lib/simplegit.rb |    5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    first commit

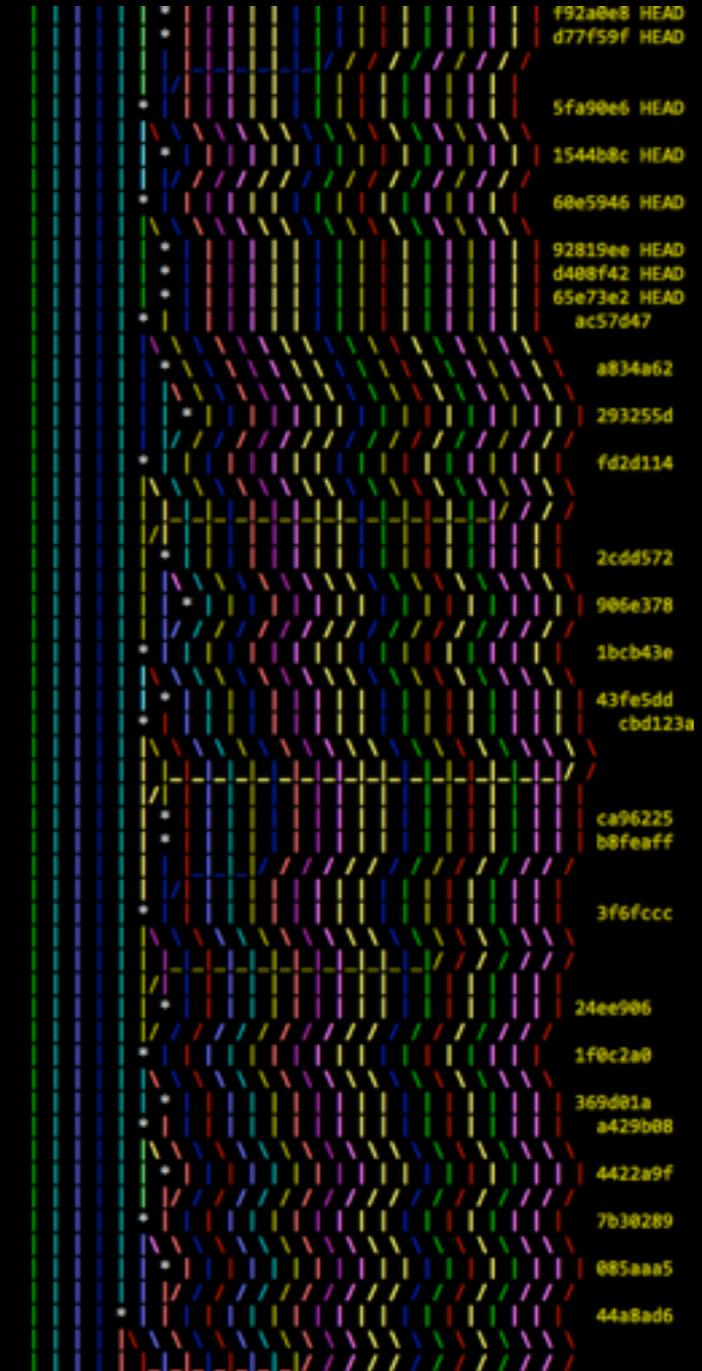
README          |    6 ++++++
Rakefile        |   23 ++++++
lib/simplegit.rb |   25 ++++++
3 files changed, 54 insertions(+)
```

# git log hacking

```
$ git log --pretty=oneline
ca82a6dff817ec66f44342007202690a93763949 changed the version number
085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7 removed unnecessary test code
allbef06a3f659402fe7563abf99ad00de2209e6 first commit
```

```
$ git log --pretty=format:"%h - %an, %ar : %s"
ca82a6d - Scott Chacon, 11 months ago : changed the version number
085bb3b - Scott Chacon, 11 months ago : removed unnecessary test code
allbef0 - Scott Chacon, 11 months ago : first commit
```

```
$ git log --pretty=format:"%h %s" --graph
* 2d3acf9 ignore errors from SIGCHLD on trap
* 5e3ee11 Merge branch 'master' of git://github.com/dustin/grit
|
| * 420eac9 Added a method for getting the current branch.
* | 30e367c timeout code and tests
* | 5a09431 add timeout protection to grit
* | e1193f8 support for heads with slashes in them
|/
* d6016bc require time for xmlschema
* 11d191e Merge branch 'defunkt' into local
```



# git remote

```
$ git remote -v  
origin  git://github.com/schacon/ticgit.git (fetch)  
origin  git://github.com/schacon/ticgit.git (push)
```

# git pull

```
git pull.
```

```
~/dev/luigi (master) $ git pull --rebase origin master
From https://github.com/rantav/luigi
 * branch          master          -> FETCH_HEAD
First, rewinding head to replay your work on top of it...
Applying: Add String column types in the history db a length -
Applying: Minor refactoring of Worker to allow easier subclass
Applying: - don't disable existing loggers when logging.config
Applying: Add ON DUPLICATE KEY UPDATE clause to the MySQL target
```

```
~/dev/luigi (master) $ git pl
Current branch master is up to date.
```

# To rebase or not to rebase?

- Prefer rebase
  - this keeps the commit log cleaner
  - `git pl`
- Sometimes an actual merge is needed
  - So don't rebase
  - For example, when merging branches

# git push

```
$ git push origin master
```

```
~/dev/luigi (master) $ git push  
Counting objects: 24, done.  
Delta compression using up to 8 threads.  
Compressing objects: 100% (17/17), done.  
Writing objects: 100% (17/17), 1.94 KiB |  
Total 17 (delta 13), reused 0 (delta 0)  
To https://github.com/rantav/luigi.git  
d046ae5..65d9388 master -> master
```

A black and white photograph of a woman in a white dress floating horizontally in dark water. The woman is positioned in the upper half of the frame, with her body parallel to the water's surface. The text "SVN Parallels" is overlaid in large, white, sans-serif font across the middle of the image, partially obscuring the woman's lower body and the water below. The overall mood is serene and ethereal.

# SVN Parallels

# Git for SVN

```
git clone url  
git pull
```

```
svn checkout url  
svn update
```

```
git diff
```

```
svn diff | less
```

```
git diff rev path
```

```
svn diff -rrev path
```

```
git apply
```

```
patch -p0
```

```
git status
```

```
svn status
```

```
git checkout path
```

```
svn revert path
```

# Git for SVN

```
git add file  
git rm file  
git mv file
```

```
svn add file  
svn rm file  
svn mv file
```

```
git commit -a
```

```
svn commit
```

```
git log  
git blame file
```

```
svn log | less  
svn blame file
```

```
git show rev:path/to/file  
git show rev:path/to/directory  
git show rev
```

```
svn cat url  
svn list url  
svn log -rrev url  
svn diff -crev url
```

# Git for SVN

```
git pull
```

```
svn update
```

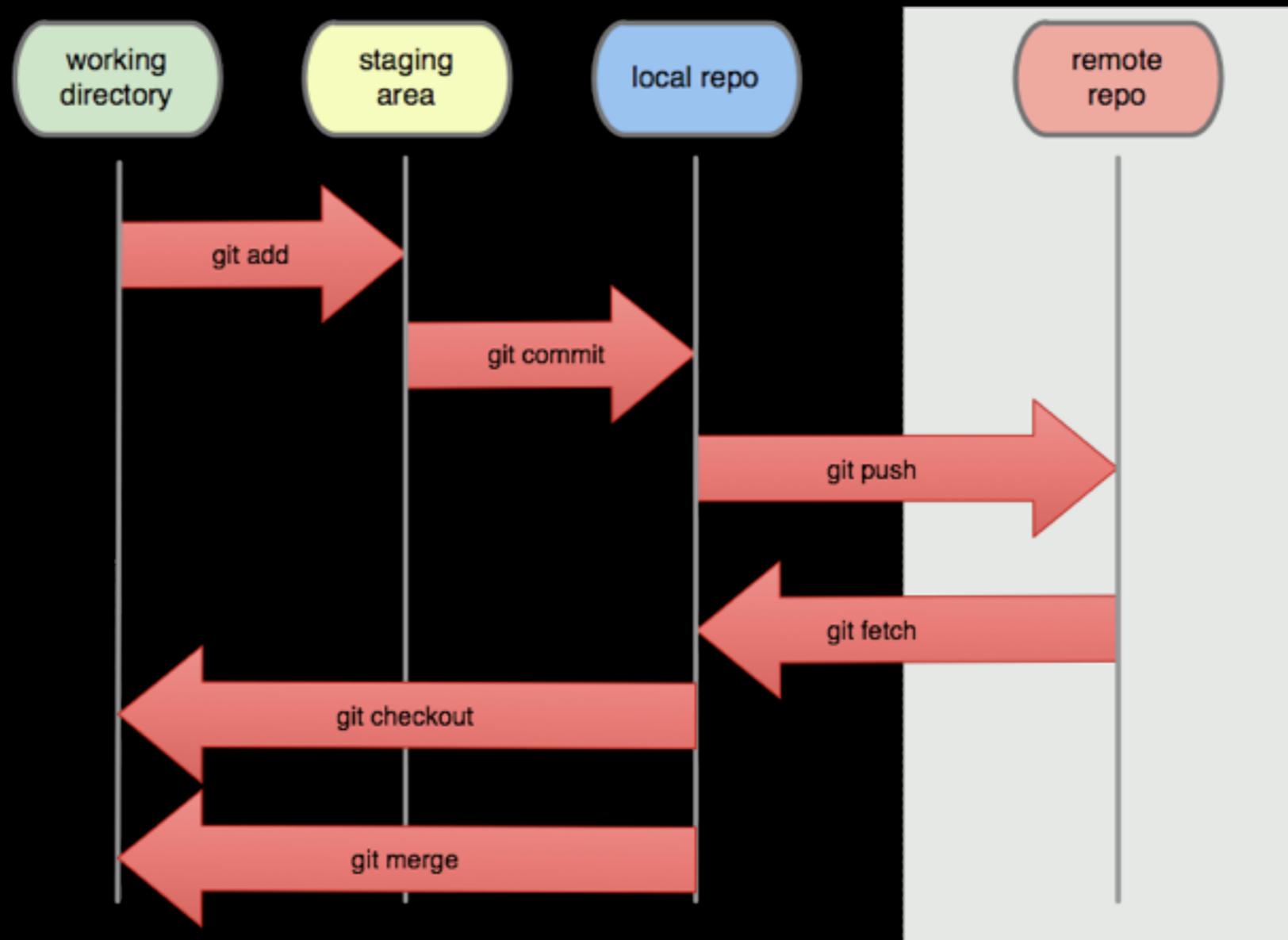
```
git clone url
```

```
svn checkout url
```

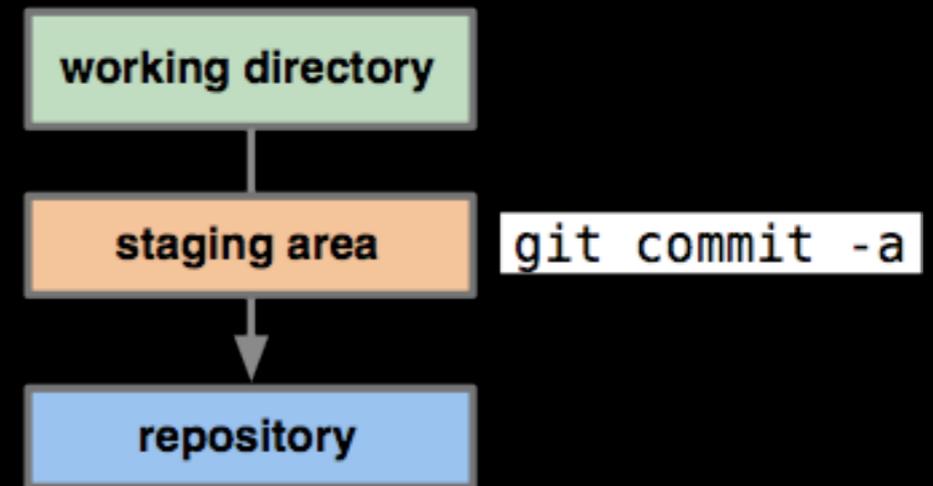
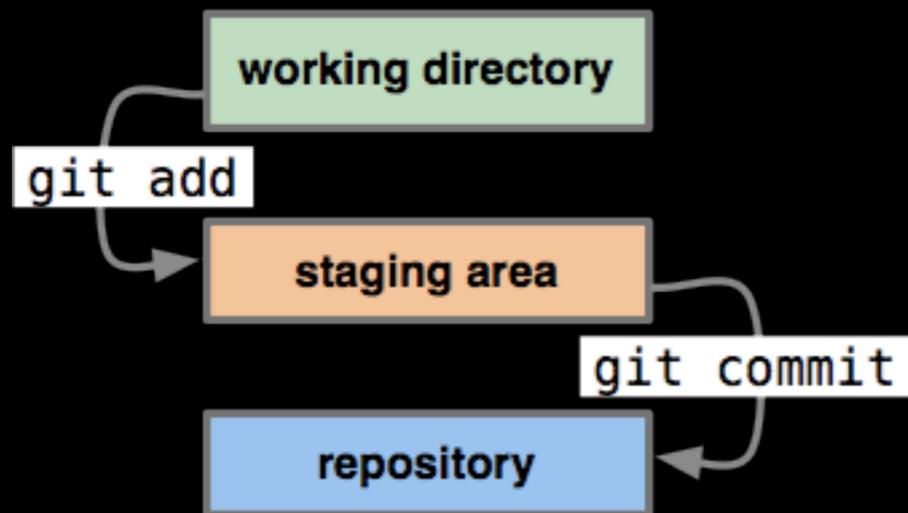
A man is shown from the chest up, submerged in dark water. He is looking intently at a glowing, spherical object he is holding in his hands. The water is dark, and the light from the orb illuminates his face and hands, creating a dramatic, ethereal atmosphere. The overall tone is mysterious and focused.

# Git Lifecycle

# Git Lifecycle



# The Staging Area (AKA The Index)

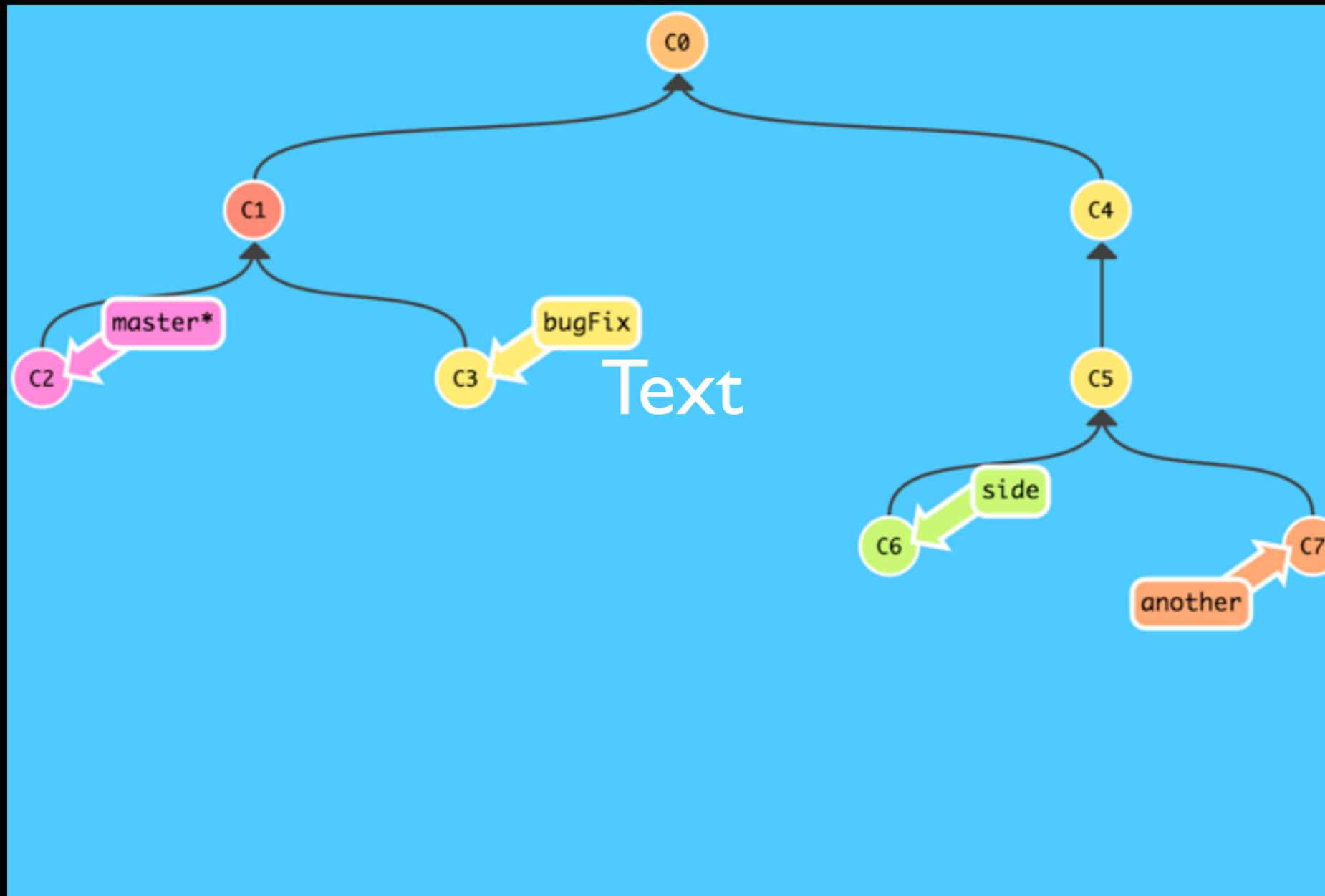


```
$ git diff --cached
diff --git a/README b/README
new file mode 100644
index 0000000..03902a1
--- /dev/null
+++ b/README2
@@ -0,0 +1,5 @@
+grit
+ by Tom Preston-Werner, Chris Wanstrath
+ http://github.com/mojombo/grit
+
+Grit is a Ruby library for extracting inform
```



# Branching

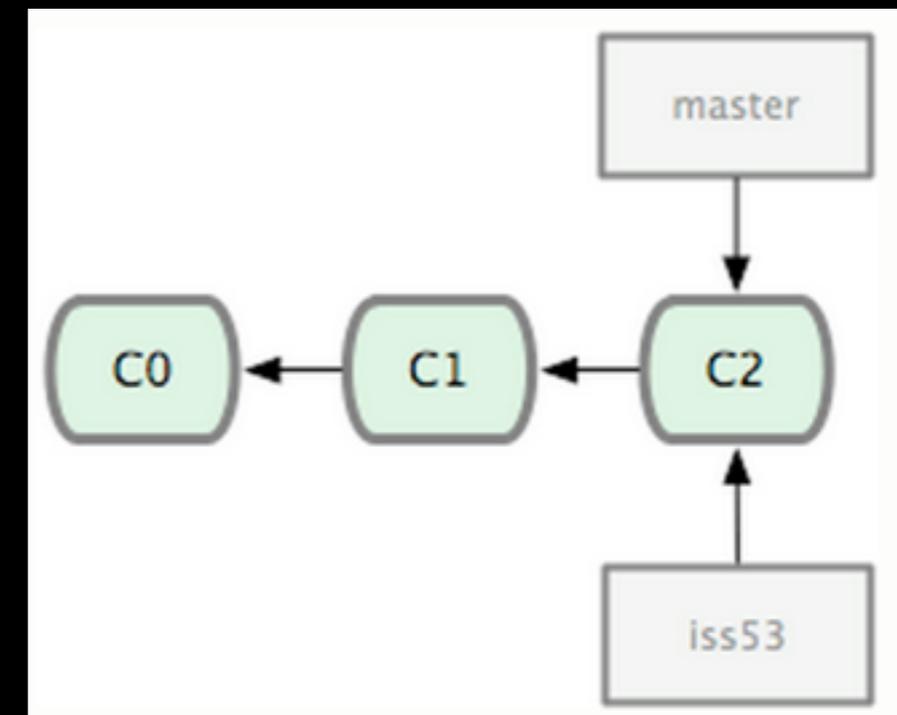
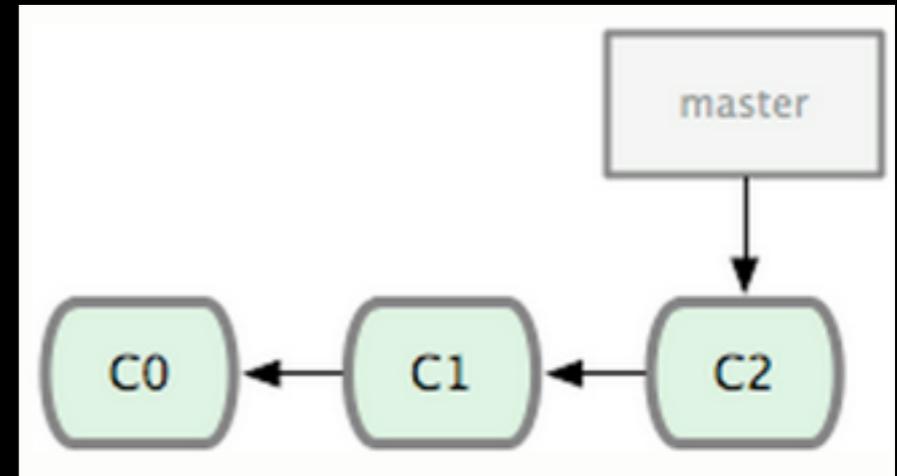
# A cool demo



<http://pcottle.github.io/learnGitBranching/?demo>

# Branching

```
$ git checkout -b iss53  
Switched to a new branch 'iss53'
```

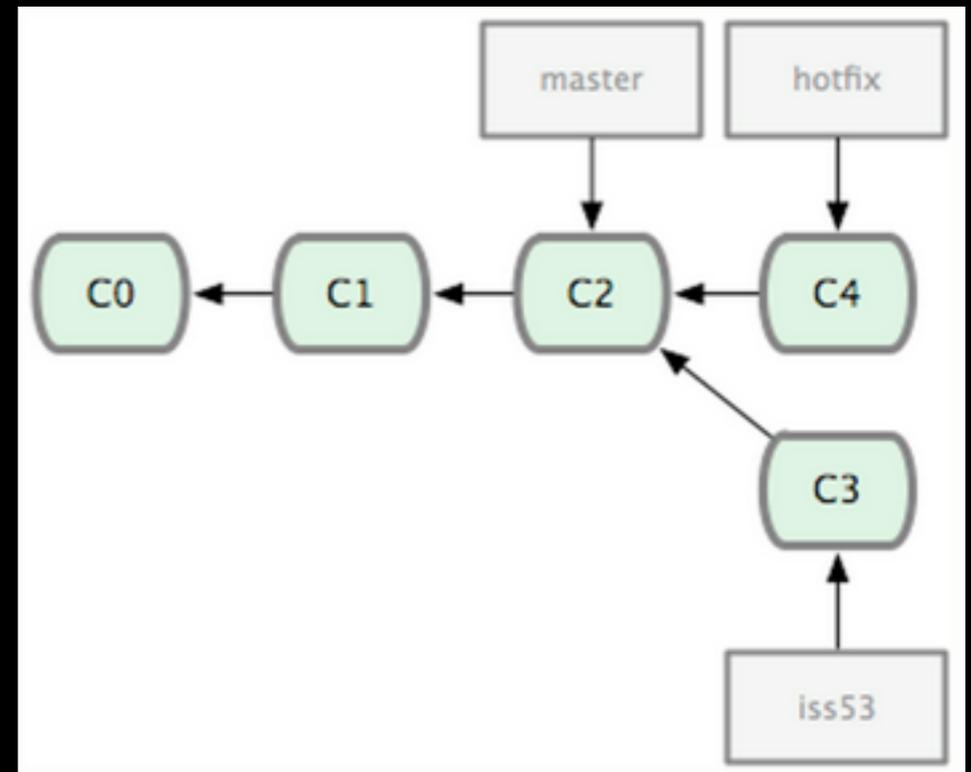
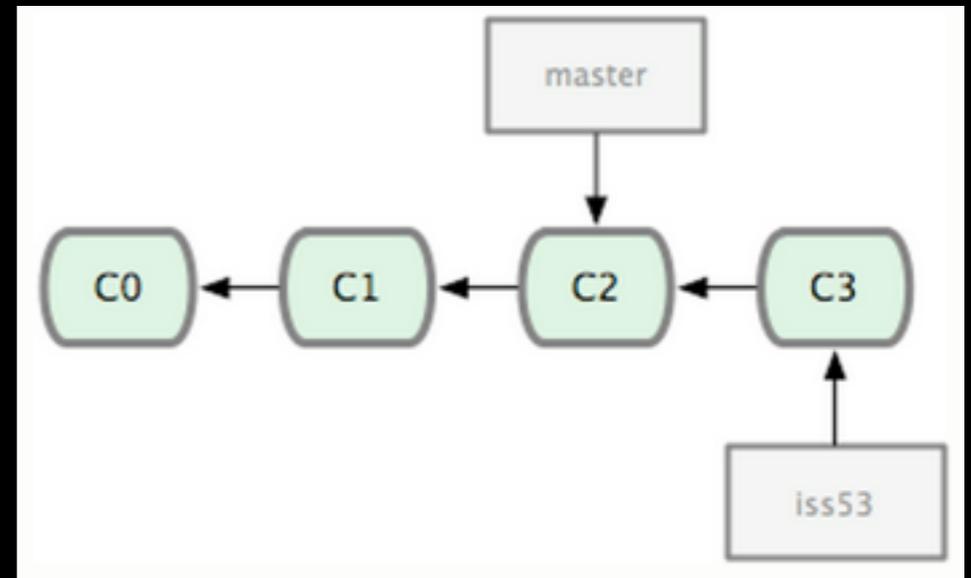


# Branching

```
$ vim index.html  
$ git commit -a -m 'added a new footer [issue 53]'
```

```
$ git checkout master  
Switched to branch 'master'
```

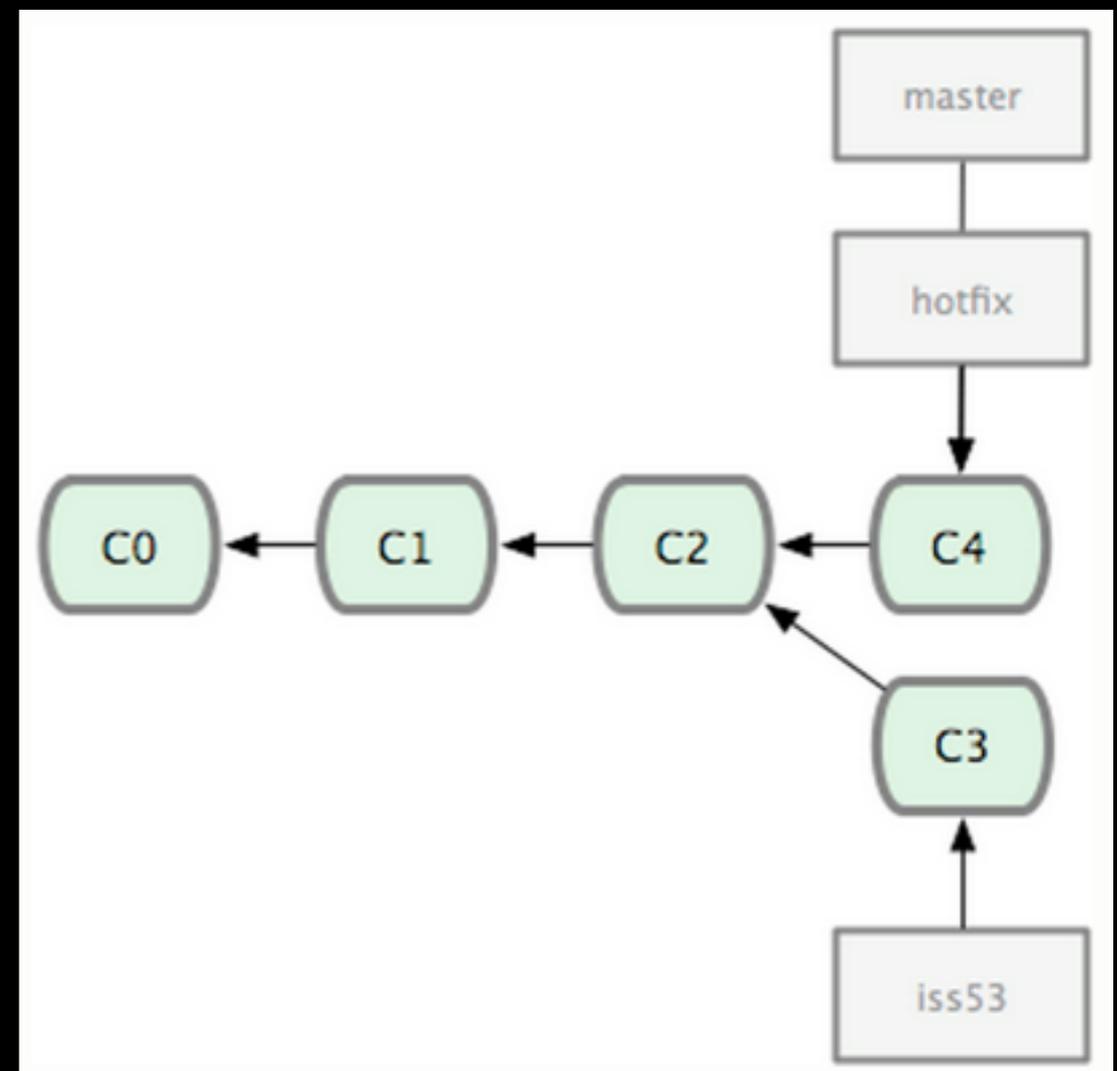
```
$ git checkout -b hotfix  
Switched to a new branch 'hotfix'  
$ vim index.html  
$ git commit -a -m 'fixed the broken email address'  
[hotfix 3a0874c] fixed the broken email address  
1 files changed, 1 deletion(-)
```



# Branching - Merging

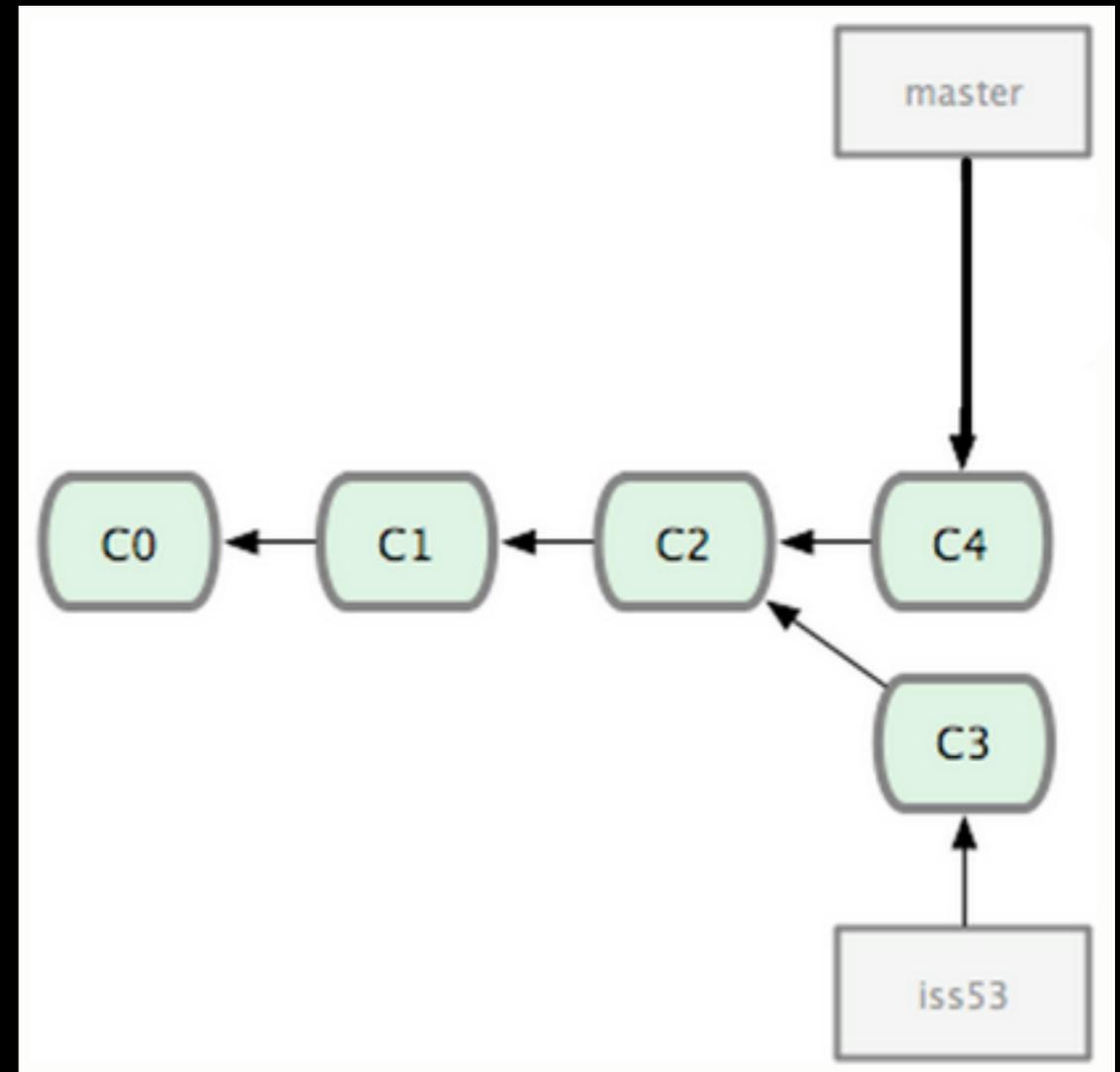
(fast forward)

```
$ git checkout master
$ git merge hotfix
Updating f42c576..3a0874c
Fast-forward
 README | 1 -
 1 file changed, 1 deletion(-)
```



# Branching - Deleting

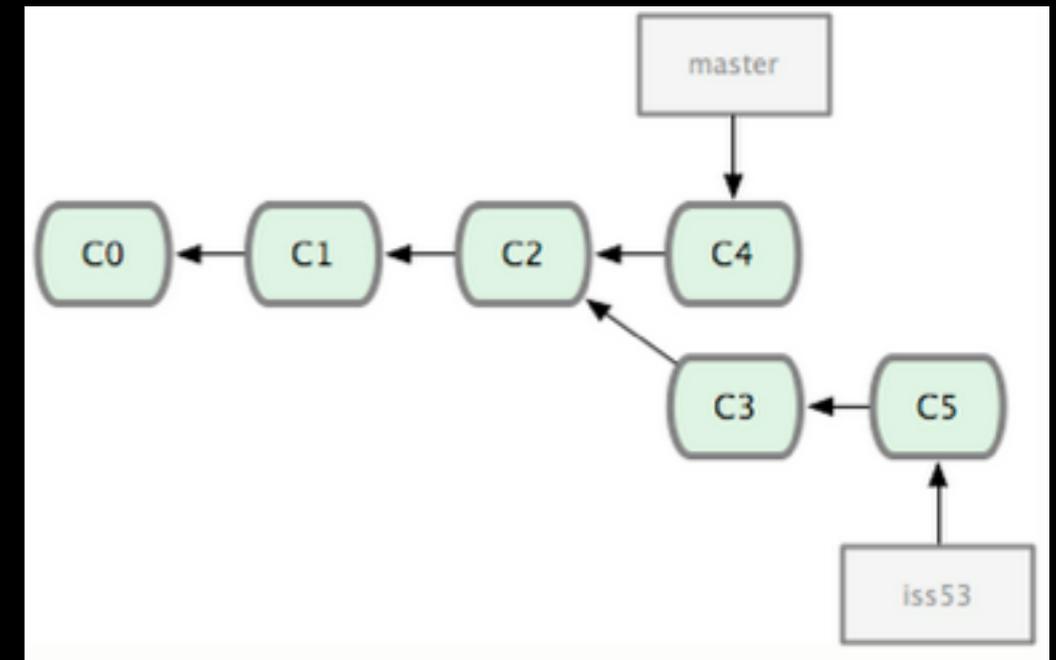
```
$ git branch -d hotfix  
Deleted branch hotfix (was 3a0874c).
```



# Branching

cont

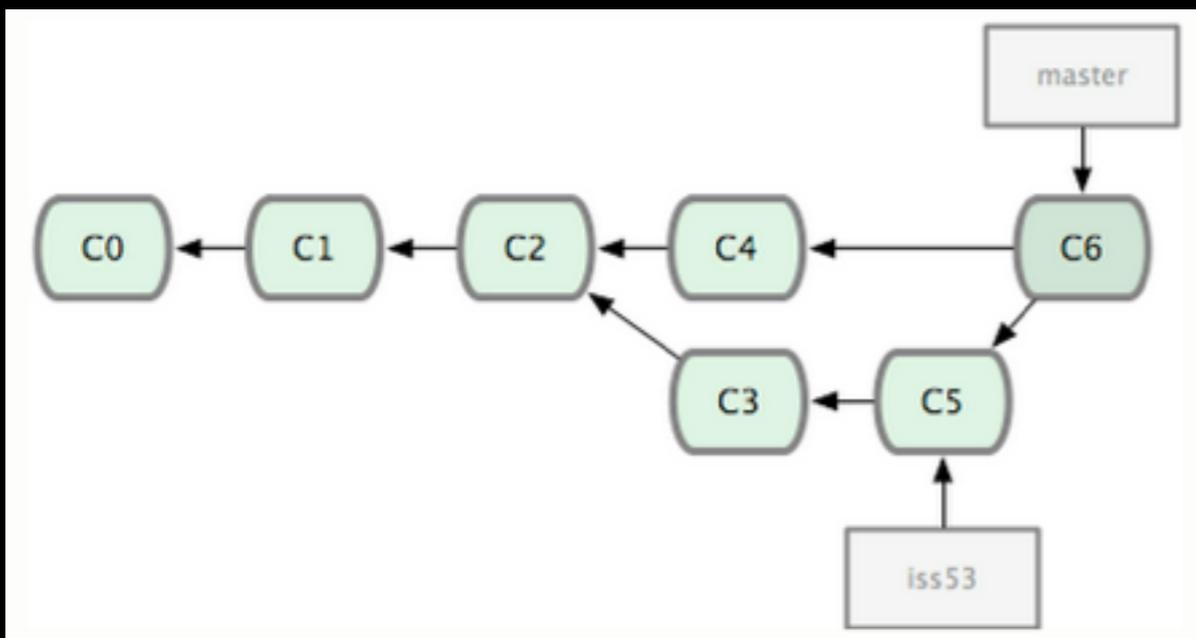
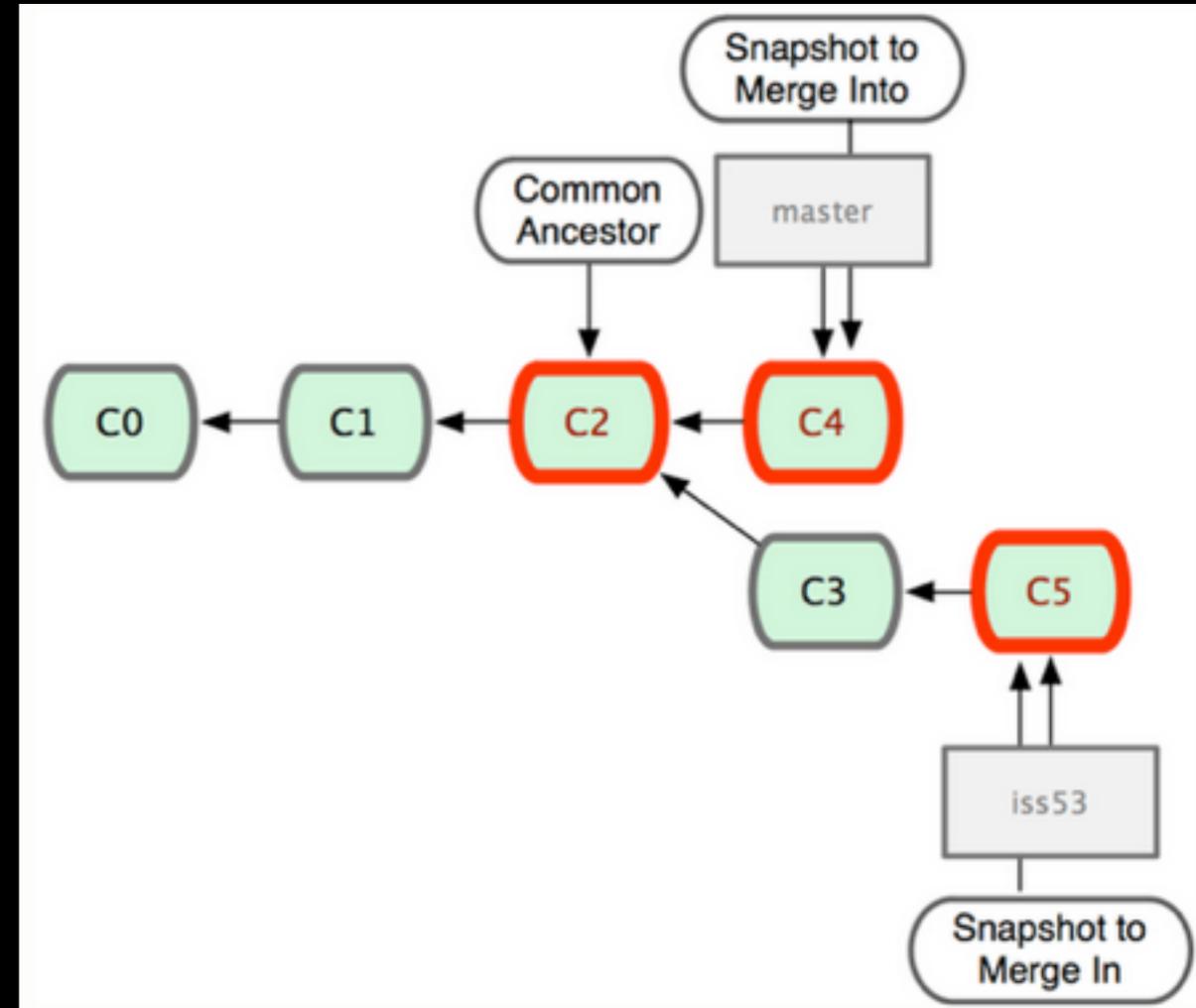
```
$ git checkout iss53  
Switched to branch 'iss53'  
$ vim index.html  
$ git commit -a -m 'finished the new footer [issue 53]'  
[iss53 ad82d7a] finished the new footer [issue 53]  
1 file changed, 1 insertion(+)
```



# Branching - Merging

(3 way merge)

```
$ git checkout master
$ git merge iss53
Auto-merging README
Merge made by the 'recursive' strategy.
 README | 1 +
 1 file changed, 1 insertion(+)
```



# Branching - Management

```
$ git branch  
  iss53  
* master  
  testing
```

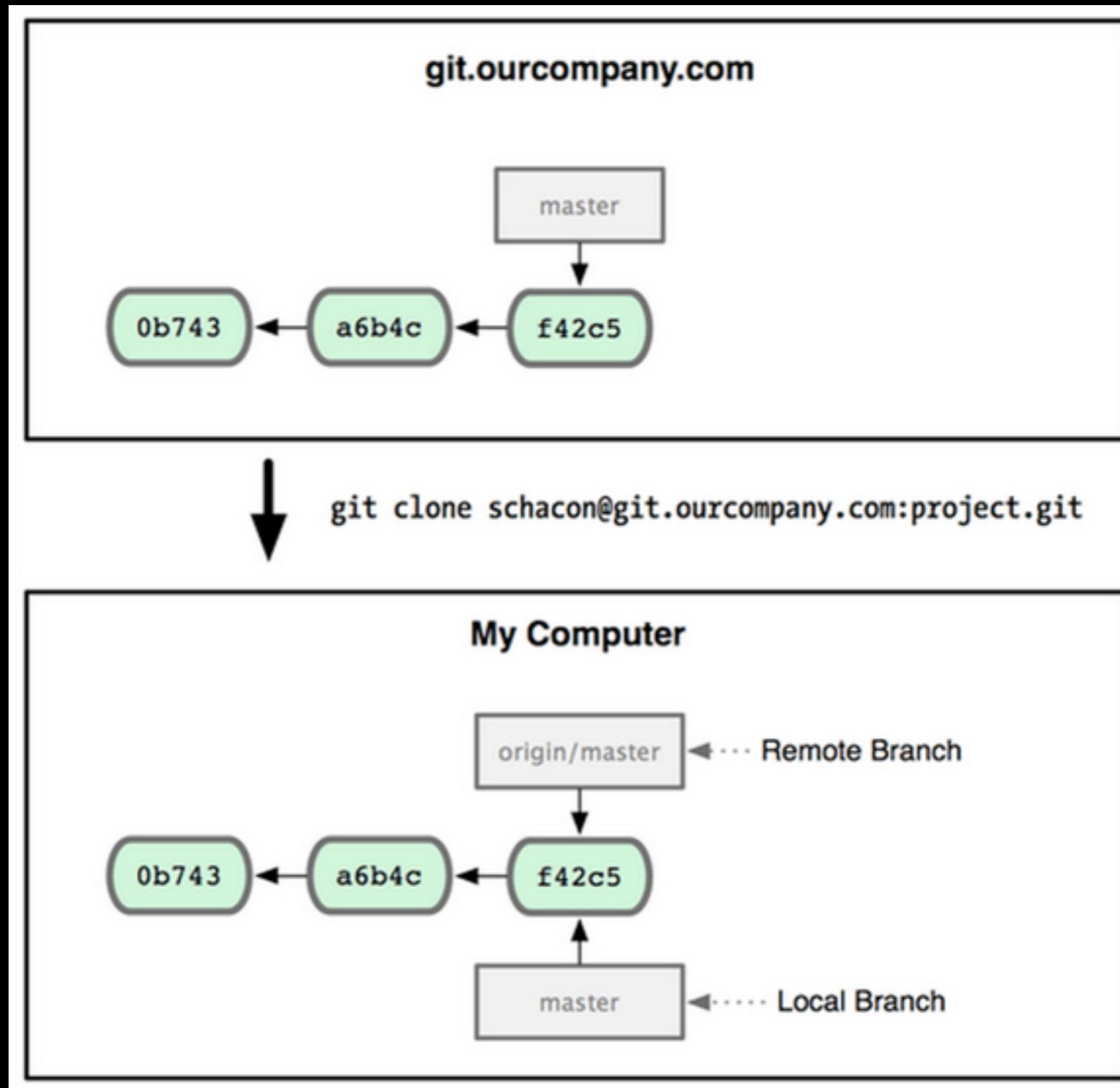
```
$ git branch -v  
  iss53    93b412c fix javascript issue  
* master  7a98805 Merge branch 'iss53'  
  testing 782fd34 add scott to the author list in the readmes
```

```
$ git branch --merged  
  iss53  
* master
```

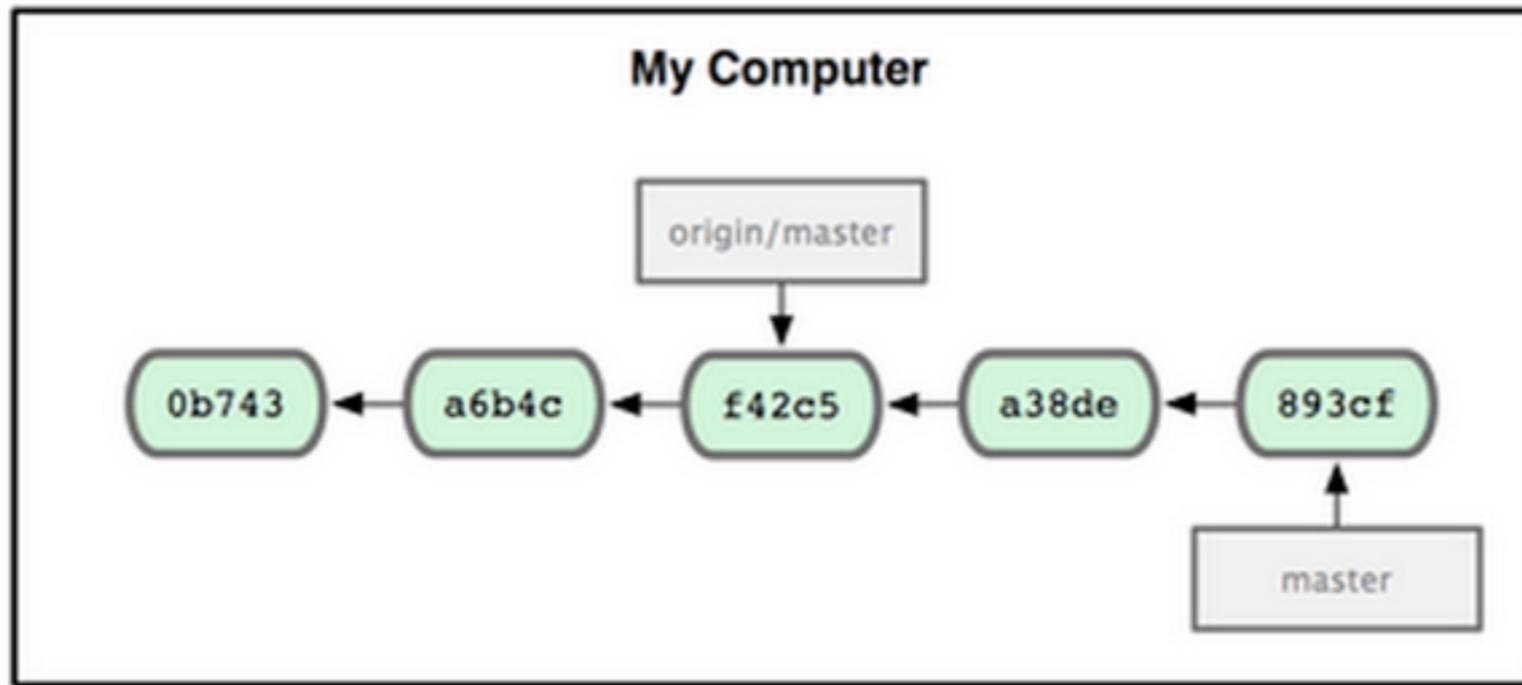
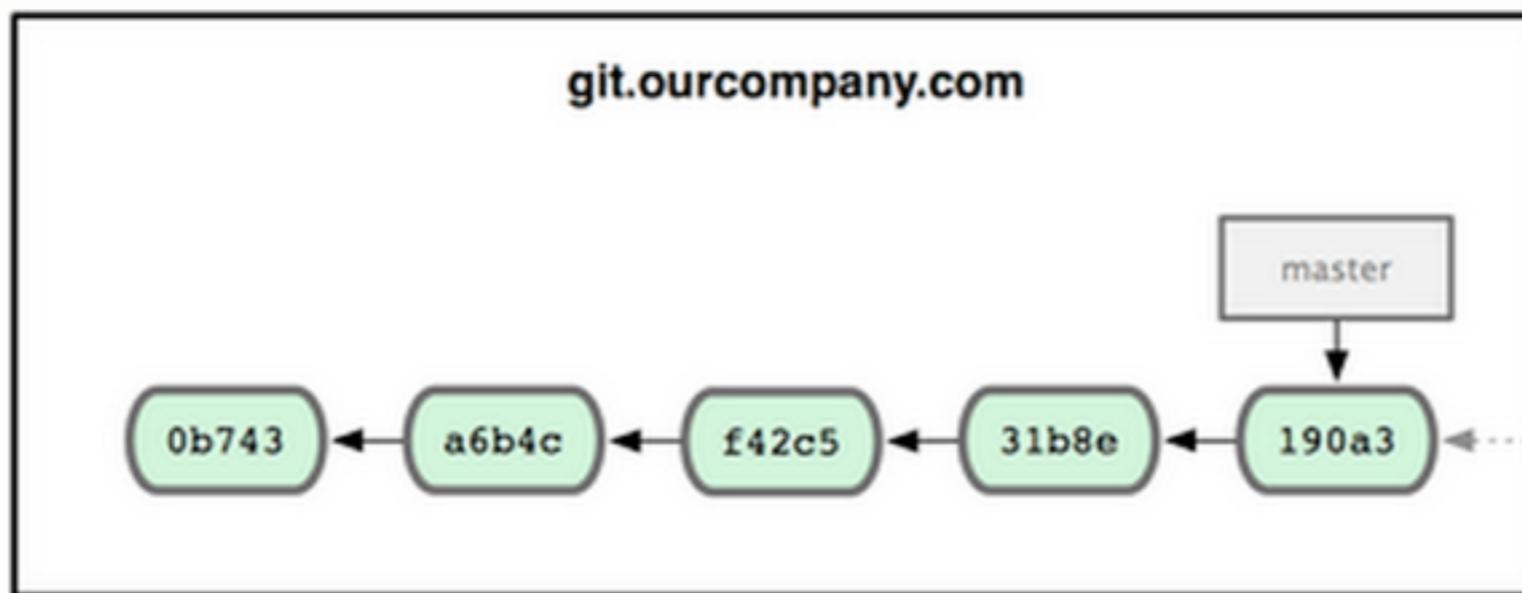
```
$ git branch --no-merged  
  testing
```

```
$ git branch -d testing  
error: The branch 'testing' is not fully merged.  
If you are sure you want to delete it, run 'git branch -D testing'.
```

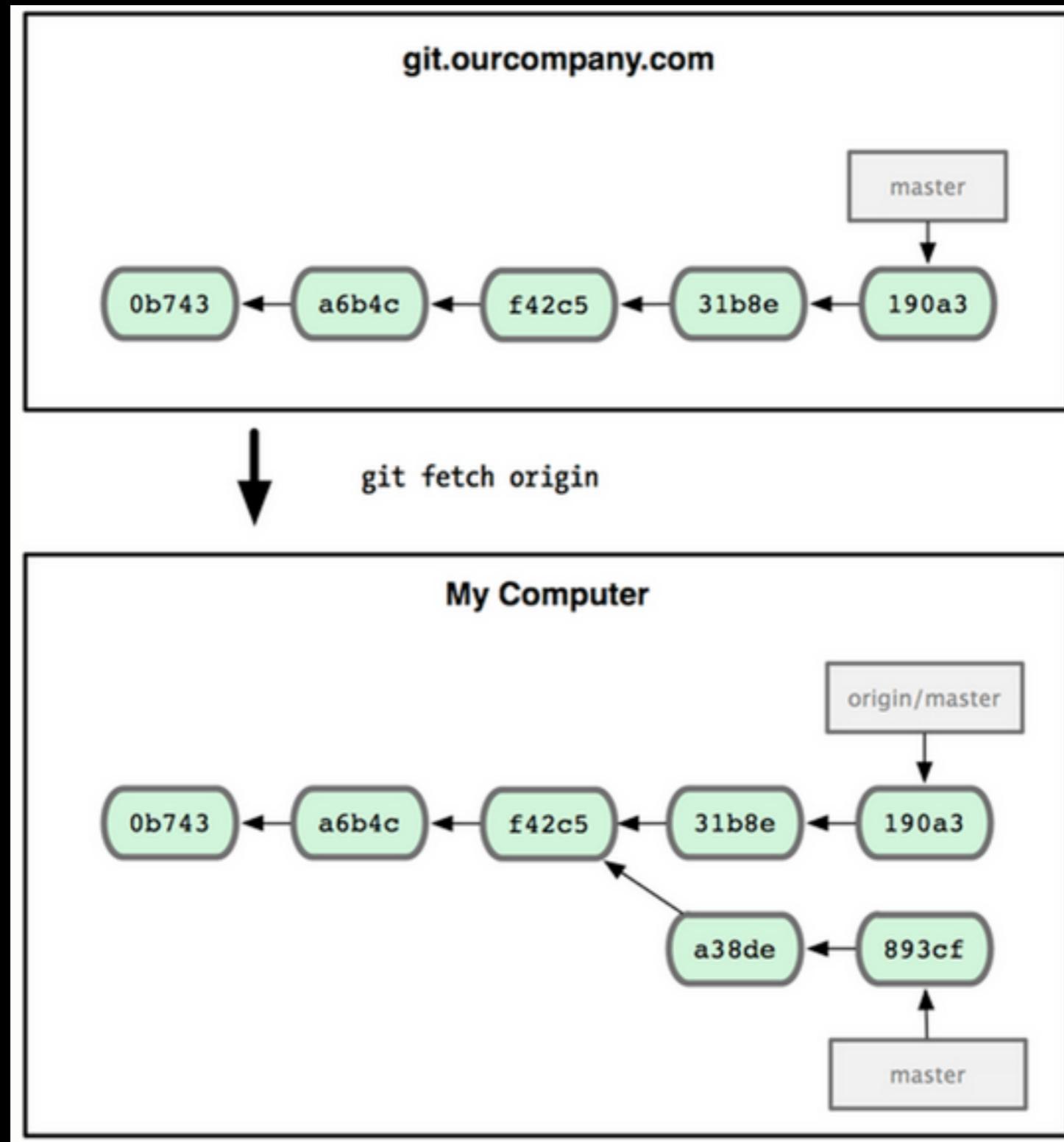
# Branching - Remote



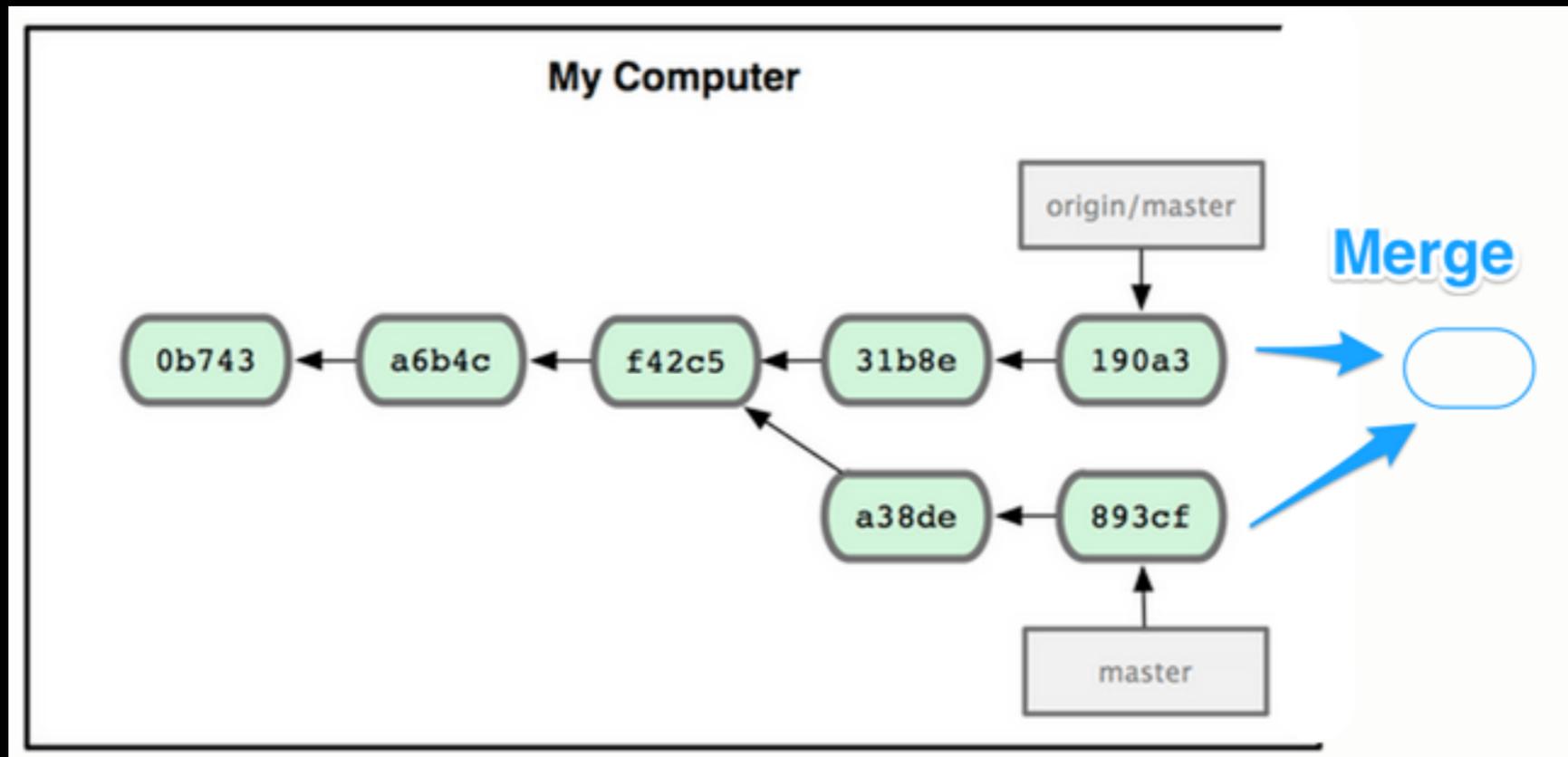
# Branching - Remote



# Branching - Remote



# Branching - Remote



`git pull == git fetch && git merge`

# Branching - Pushing

```
$ git push origin serverfix
Counting objects: 20, done.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (15/15), 1.74 KiB, done.
Total 15 (delta 5), reused 0 (delta 0)
To git@github.com:schacon/simplegit.git
* [new branch]      serverfix -> serverfix
```

```
git push
```

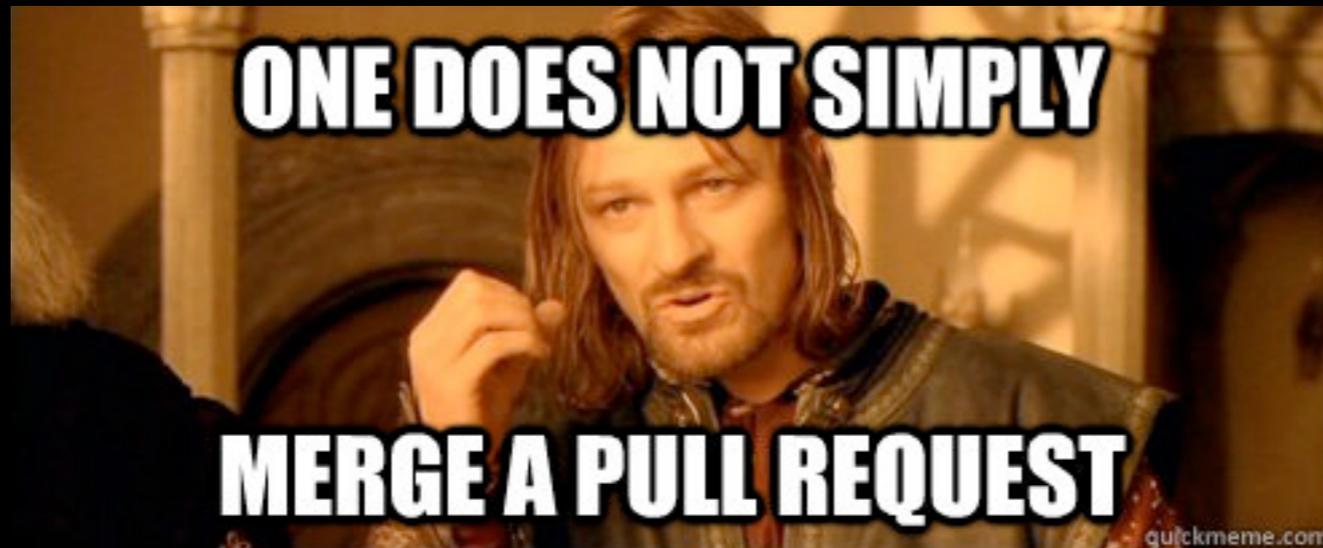
Requires tracking

```
git push origin serverfix:awesomebranch
```

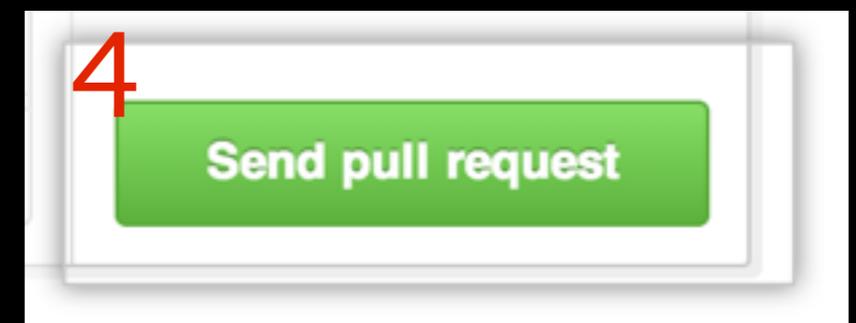
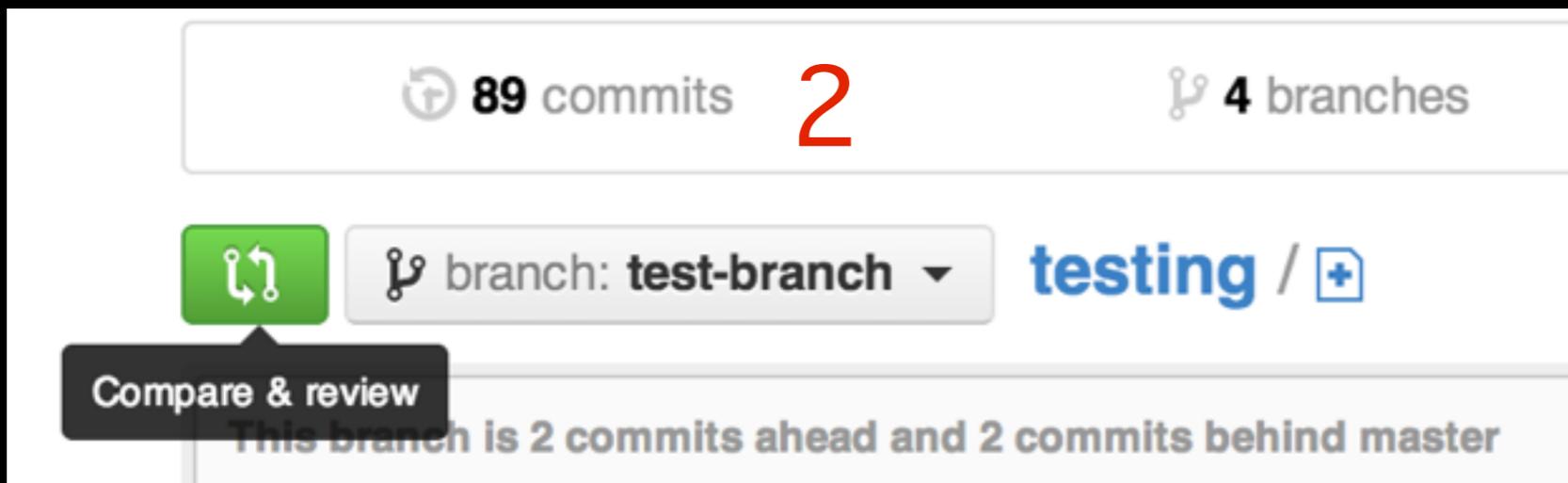
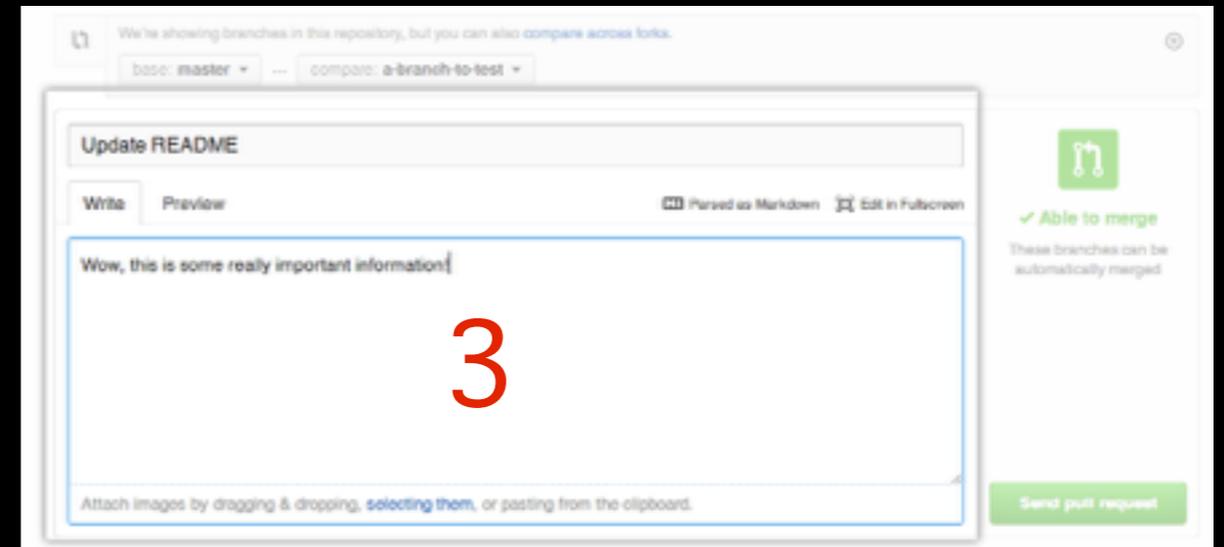
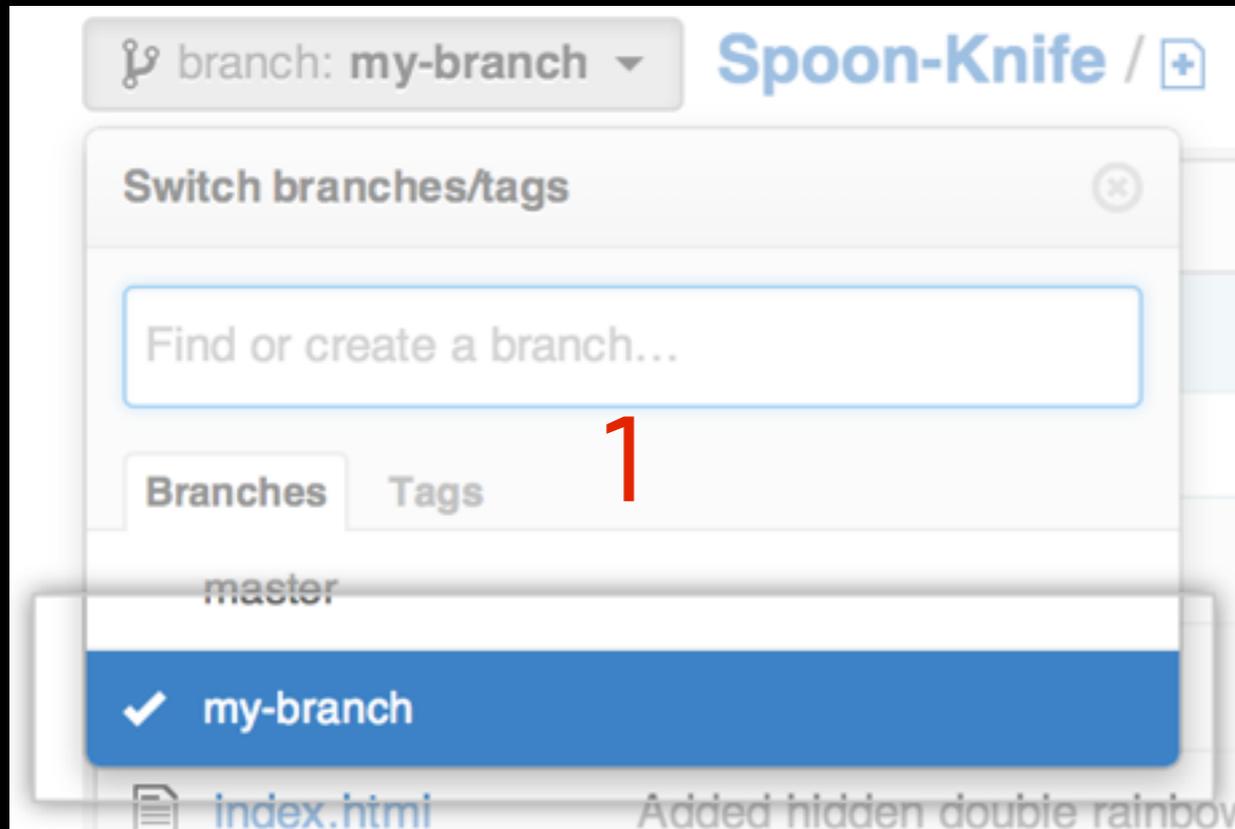
# Deleting Remote Branches

```
$ git push origin :serverfix  
To git@github.com:schacon/simplegit.git  
- [deleted]          serverfix
```

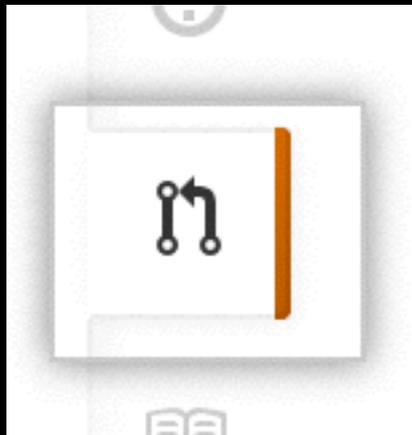
# Pull Requests



# Creating a Pull Request



# Reviewing a Pull Request



Discussion   **Commits** 1   Files Changed 1

Showing 1 unique commit by 1 author.

Jan 30, 2013

 codercat   Update README 824e3f0

Discussion   Commits 1   **Files Changed** 1

Showing 1 changed file with 1 addition and 1 deletion. Show Diff Stats

2 █ █ README View file @ 824e3f0

```
...  ...  @@ -1,1 @@
1      -All that's missing is the fork. Heh.
      \ No newline at end of file
1      +All that's missing is the fork. Heh!
```

 38 58   **Tip:** You can add notes to lines in a file.  
39 59   Hover to the left of a line to make a note  
40 60

# Merging Pull Request Online

 **This pull request can be automatically merged.**  
You can also merge branches on the [command line](#).

 Merge pull request

 **Merge pull request #2 from bernars/test-branch**

Create contributing.md

 **bernars**  
bernars@github.com

# Merging Pull Requests Offline

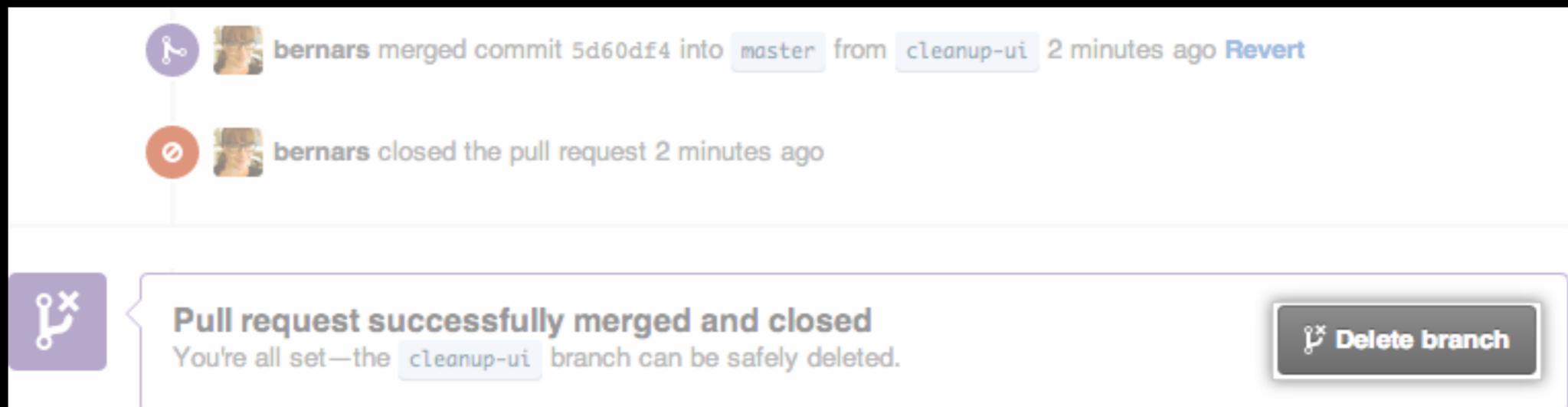
```
$ git checkout master
```

```
$ git pull https://github.com/octocat/repo.git branchname
```

- Resolve Conflicts
- Test
- Improve

```
$ git push origin master
```

# Deleting Branch After PR Merge



Activity feed showing:

- bernars merged commit 5d60df4 into `master` from `cleanup-ui` 2 minutes ago [Revert](#)
- bernars closed the pull request 2 minutes ago

Success message:

**Pull request successfully merged and closed**  
You're all set—the `cleanup-ui` branch can be safely deleted.

[Delete branch](#)

The background features a series of overlapping, wavy lines that create a sense of motion and flow. The lines are primarily orange and yellow on the left side, transitioning to a teal and blue on the right side. The overall effect is reminiscent of a digital or data stream.

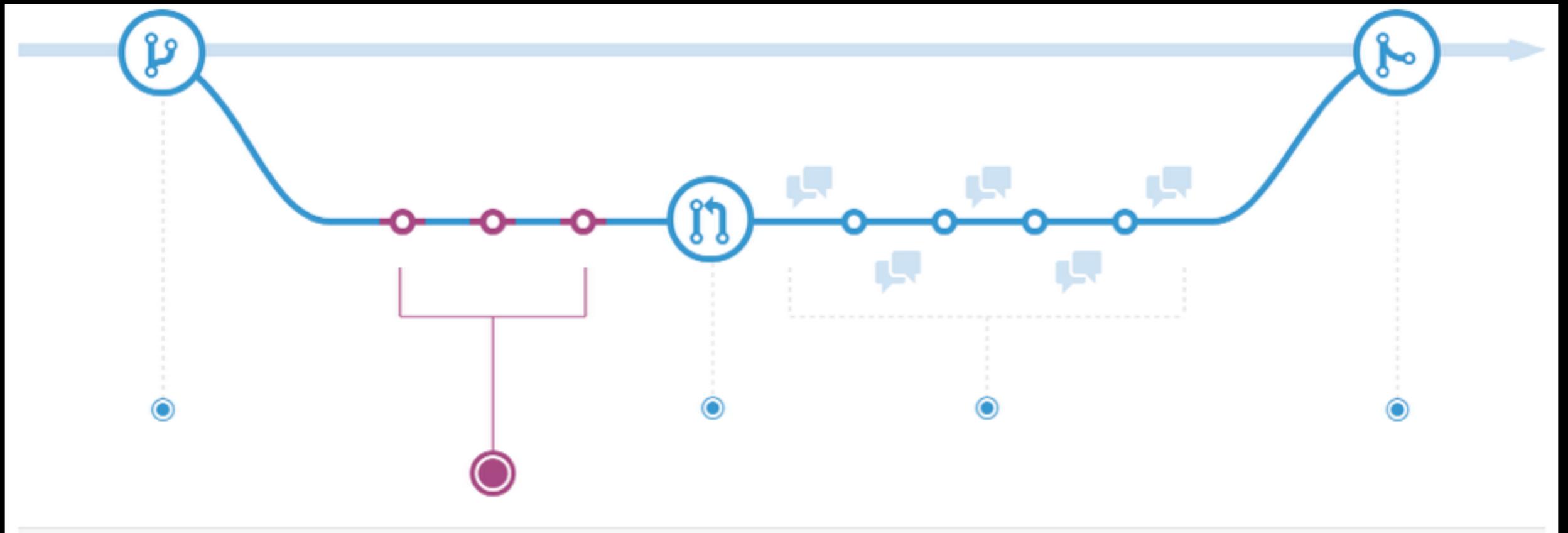
# Git Flow



**KEEP  
CALM  
AND  
GO WITH  
THE FLOW**

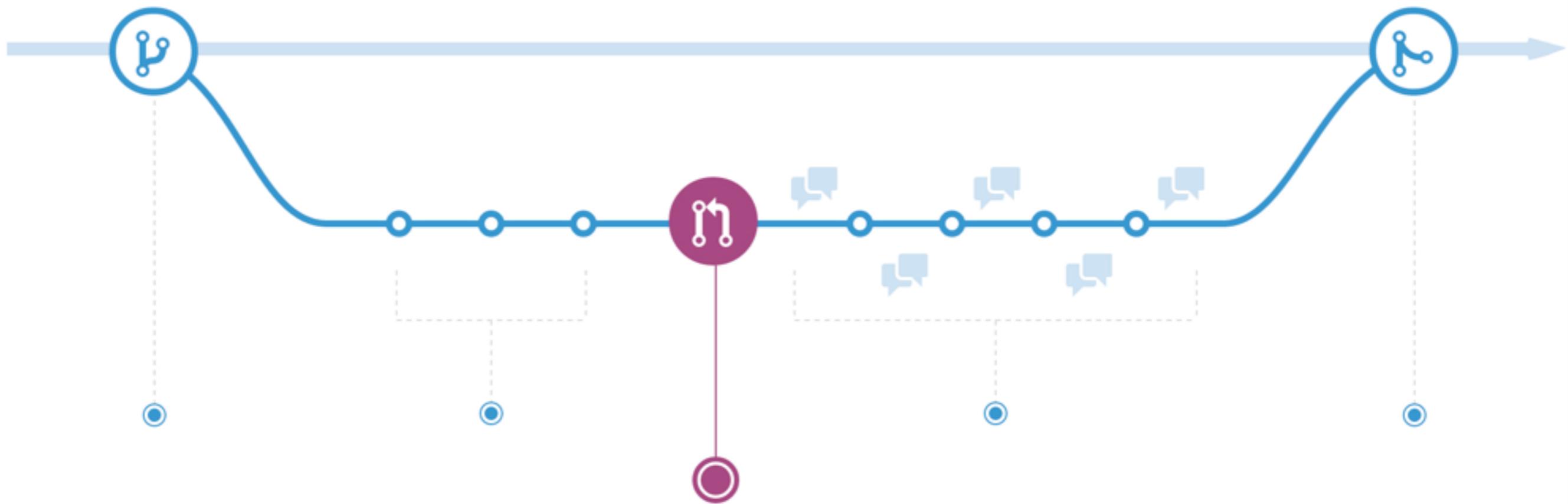


# 2. Add Commits



Commit messages are super important!!!

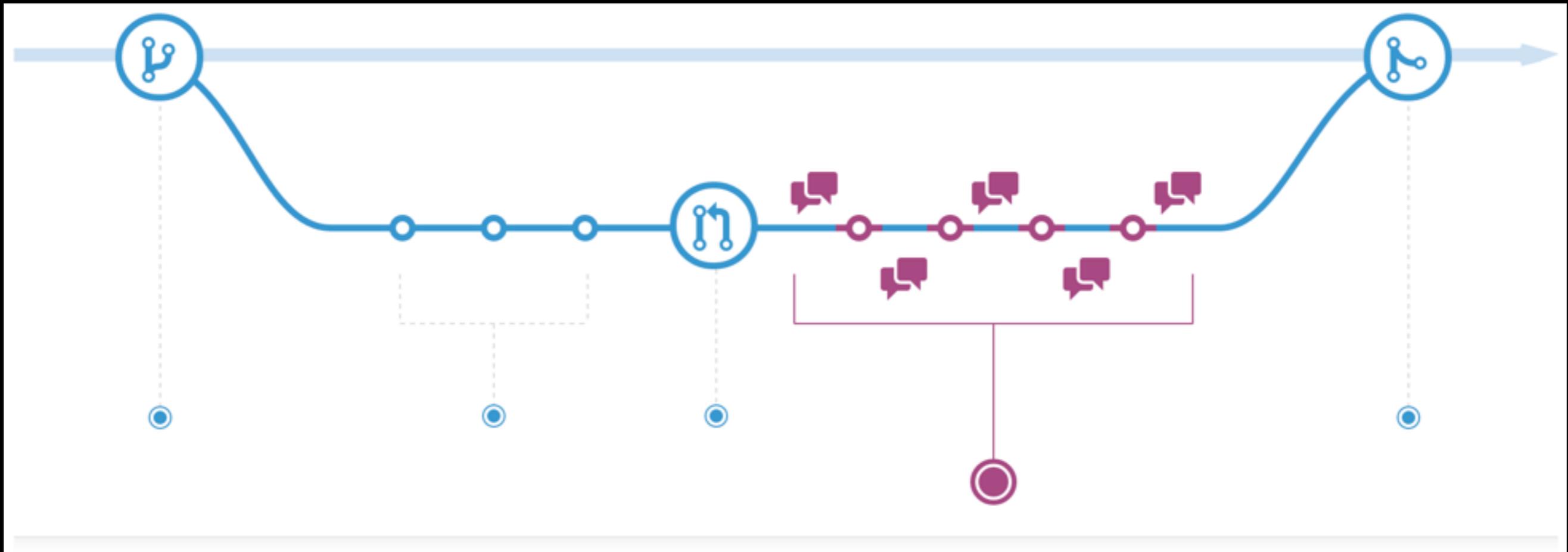
# 3. Open a Pull Request



Make sure you assign a reviewer!

# 4. Discuss

AKA Code Review



Continue to commit and push.  
This will be reflected

# 5. Merge

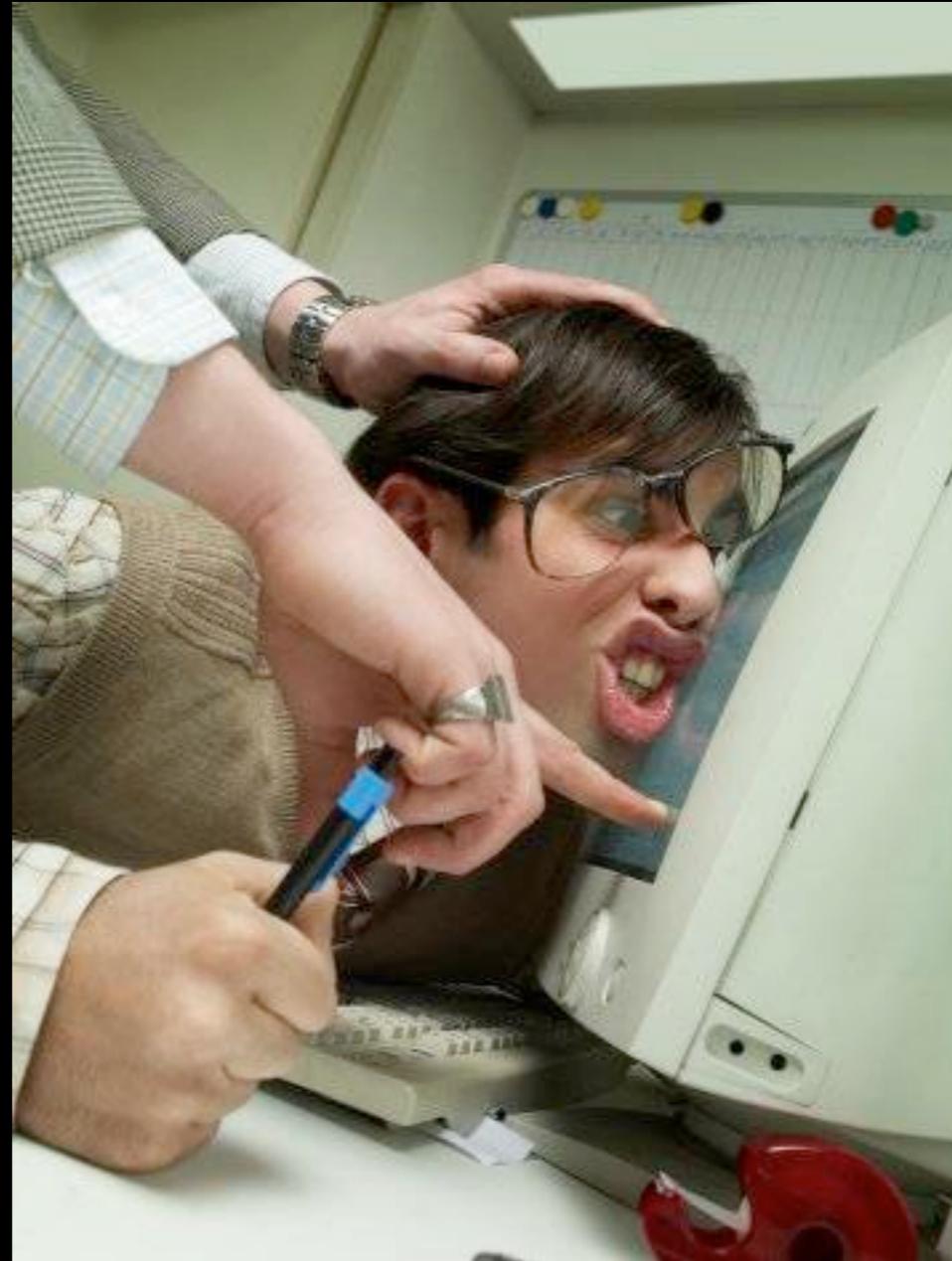


Merge means: It's good for production.  
**master** is always deployable

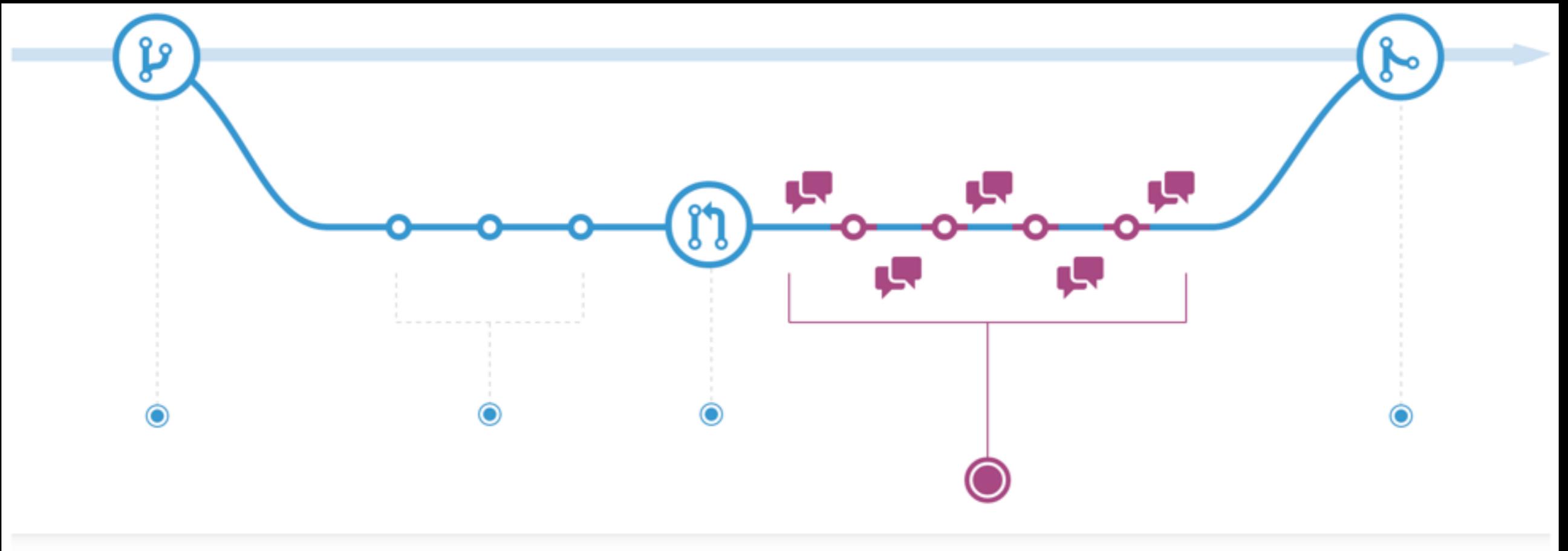
# Pull Request Demo

**D E M O**

# Code Review



# Code Review





# Protips

# Protip: How to transition your team from svn to git

1. Use git-svn, or Svn2git
2. Start using git “as svn” eg no branches
3. Start using branches and pull requests (if github)
4. Personal opinion: Use only CLI tools



# Protip: `git rebase -i`

```
pick 661484b Fix layout and a typo
pick 44afeaf Insert the logo back, to the index page. It doesn't look good without
pick 28f804a Fix in-page navigation
pick 94ea5bb Update the mission statement

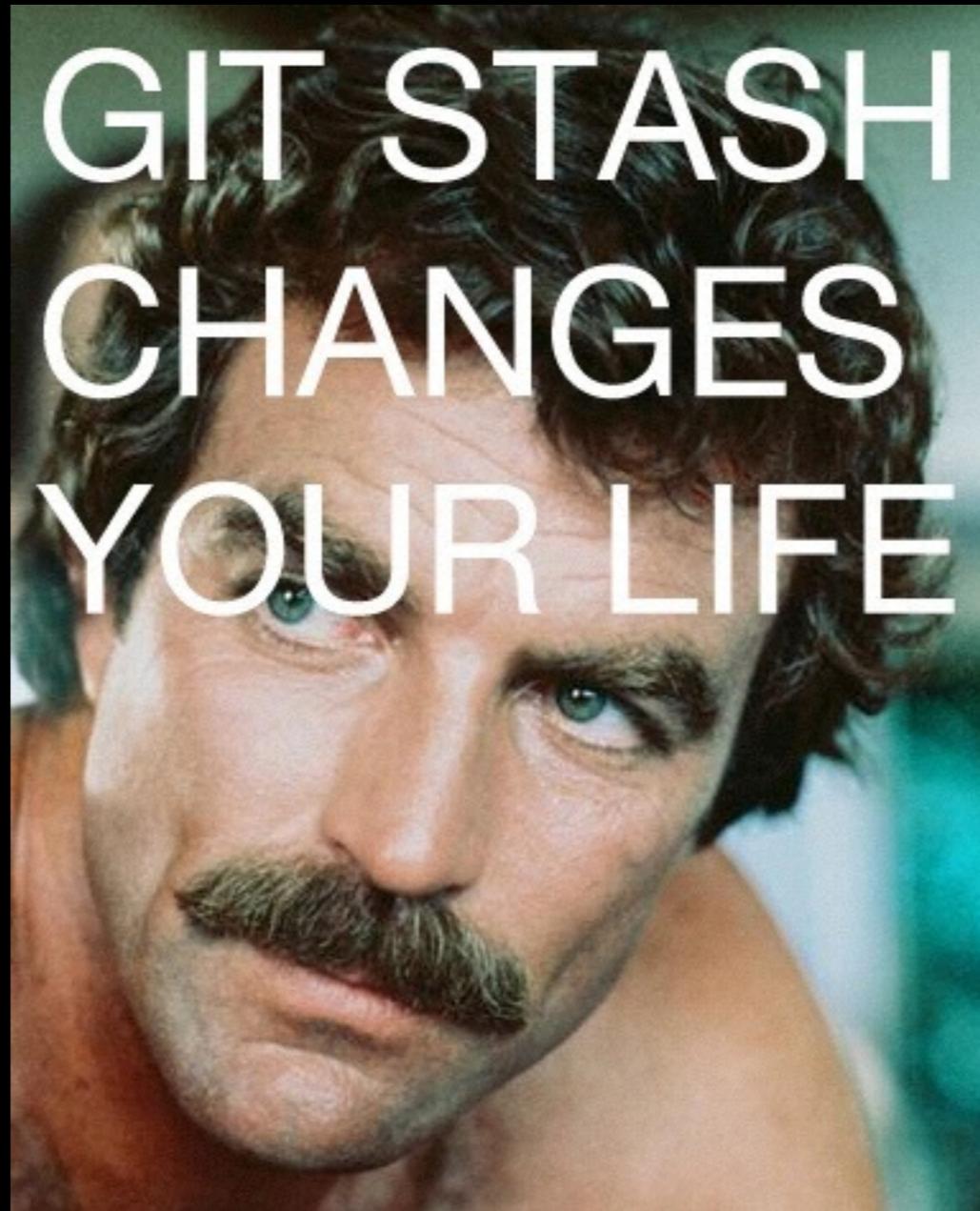
# Rebase a58f1df..94ea5bb onto a58f1df
#
# Commands:
# p, pick = use commit
# r, reword = use commit, but edit the commit message
# e, edit = use commit, but stop for amending
# s, squash = use commit, but meld into previous commit
# f, fixup = like "squash", but discard this commit's log message
# x, exec = run command (the rest of the line) using shell
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
# Note that empty commits are commented out
```

# Protip: zsh integration

```
~/dev/luigi (master) $ git l  
local-fetch  -- duplicate another git repository on a local system  
log          -- show commit logs  
lost-found   -- recover lost references that luckily have not yet  
ls-files     -- information about files in the index/working dire  
ls-remote    -- show references in a remote or local repository  
ls-tree      -- display tree object in human-readable form
```



**Protip: `git stash`**



A woman with long, wavy hair is submerged in dark water, wearing a white, flowing dress. Her reflection is visible on the water's surface above her. The scene is dimly lit, with light reflecting off the water and the dress. The word "Extra" is overlaid in white text in the center of the image.

**Extra**

# Conflicts



## Conflicts

# Edit Collision

```
$ git status
# On branch branch-b
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add ..." to mark resolution)
#
# both modified:      planets.md
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
the number of planets are
<<<<<<< HEAD
nine
=====
eight
>>>>>>> branch-a
```

```
# Edit file and then
$ git add
$ git commit
```

## Conflicts

# Removed File

```
CONFLICT (modify/delete): README.md deleted in HEAD and modified in branch-b.  
Version branch-b of README.md left in tree.  
Automatic merge failed; fix conflicts and then commit the result.
```

```
$ git status  
# On branch branch-c  
# You have unmerged paths.  
#   (fix conflicts and run "git commit")  
#  
# Unmerged paths:  
#   (use "git add/rm ..." as appropriate to mark resolution)  
#  
# deleted by us:      README.md  
#  
no changes added to commit (use "git add" and/or "git commit -a")
```

## Conflicts

# Removed File - Keep

```
$ git add README.md

$ git commit
[branch-c 9bc3b01] Merge branch 'branch-b' into branch-c

$ git show | head
commit 9bc3b0130fe0178359d51243b5b882076a12f554
Merge: 4c80a63 7e8b679
Author: tekkub <:bear:>
Date: Sat Jun 1 18:39:40 2013 -0700

    Merge branch 'branch-b' into branch-c
>
Conflicts:
    README.md
```

## Conflicts

# Removed File - Remove

```
$ git rm README.md
README.md: needs merge
rm 'README.md'
```

Looks good, so commit it with the default message:

```
$ git commit
[branch-d 6f89e49] Merge branch 'branch-c' into branch-d

$ git show | head
commit 6f89e49189ba3a2b7440fc434f351cb041b3999e
Merge: 211261b fcc1093
Author: tekkub <tekkub@bear>
Date: Sat Jun 1 18:43:01 2013 -0700

    Merge branch 'branch-c' into branch-d
>
Conflicts:
    README.md
```

Conflicts

# Git Mergetool

- Not covered here...
- <http://git-scm.com/docs/git-mergetool>

## Conflicts

# While Rebasing

```
$ git rebase develop
First, rewinding head to replay your work on top of it...
Applying: feature/login
Using index info to reconstruct a base tree...
Falling back to patching base and 3-way merge...
Auto-merging config/environment.rb
CONFLICT (content): Merge conflict in config/environment.rb
Failed to merge in the changes.
Patch failed at 0001 feature/login
```

When you have resolved this problem run "git rebase --continue".  
If you would prefer to skip this patch, instead run "git rebase --skip".  
To restore the original branch and stop rebasing run "git rebase --abort".

# Learning More

```
~ $ git help
usage: git [--version] [--help] [-c name=value]
          [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
          [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
          [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
          <command> [<args>]

The most commonly used git commands are:
  add          Add file contents to the index
  bisect       Find by binary search the change that introduced a bug
  branch       List, create, or delete branches
  checkout     Checkout a branch or paths to the working tree
  clone        Clone a repository into a new directory
  commit       Record changes to the repository
  diff         Show changes between commits, commit and working tree, etc
  fetch        Download objects and refs from another repository
  grep         Print lines matching a pattern
  init         Create an empty Git repository or reinitialize an existing one
  log          Show commit logs
  merge        Join two or more development histories together
  mv           Move or rename a file, a directory, or a symlink
  pull         Fetch from and merge with another repository or a local branch
  push         Update remote refs along with associated objects
  rebase       Forward-port local commits to the updated upstream head
  reset        Reset current HEAD to the specified state
  rm           Remove files from the working tree and from the index
  show         Show various types of objects
  status       Show the working tree status
  tag          Create, list, delete or verify a tag object signed with GPG

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
~ $ █
```

# Refs

*(or: where did I steal from?)*

<http://git-scm.com/about>

<http://thkoch2001.github.io/whygitisbetter/>

<http://www.slideshare.net/manishchaks/subversion-to-git-migration>

<http://www.git-tower.com/blog/git-for-subversion-users-cheat-sheet-detail/>

<http://git.or.cz/course/svn.html>

# Refs

<https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line>

<http://pcottle.github.io/learnGitBranching/>

<http://jeffkreeftmeijer.com/2010/the-magical-and-not-harmful-rebase/>

<https://help.github.com/articles/using-pull-requests>

<https://help.github.com/articles/creating-a-pull-request>

<http://en.wikipedia.org/wiki/>

[Coupling\\_\(computer\\_programming\)](#)