

git

stuart halloway
<http://thinkrelevance.com>

what is git?

git is *not* svn++

git is a
filesystem

the git filesystem

content-addressable data

trees

history

completely decentralized

remote synchronization

local metadata

git is a
toolkit

git git-add git-add--interactive git-am git-annotate git-apply git-archimport
git-archive git-bisect git-blame git-branch git-bundle git-cat-file git-
check-attr git-check-ref-format git-checkout git-checkout-index git-cherry
git-cherry-pick git-citool git-clean git-clone git-commit git-commit-tree
git-config git-count-objects git-cvsexportcommit git-cvsiimport git-cvsserver
git-daemon git-describe git-diff git-diff-files git-diff-index git-diff-tree
git-fast-export git-fast-import git-fetch git-fetch--tool git-fetch-pack git-
filter-branch git-fmt-merge-msg git-for-each-ref git-format-patch git-fsck
git-fsck-objects git-gc git-get-tar-commit-id git-grep git-gui git-hash-
object git-http-fetch git-http-push git-imap-send git-index-pack git-init
git-init-db git-instaweb git-log git-lost-found git-ls-files git-ls-remote
git-ls-tree git-mailinfo git-mailsplit git-merge git-merge-base git-merge-
file git-merge-index git-merge-octopus git-merge-one-file git-merge-ours git-
merge-recursive git-merge-resolve git-merge-stupid git-merge-subtree git-
merge-tree git-mergetool git-mktag git-mktree git-mv git-name-rev git-pack-
objects git-pack-redundant git-pack-refs git-parse-remote git-patch-id git-
peek-remote git-prune git-prune-packed git-pull git-push git-quiltimport git-
read-tree git-rebase git-rebase--interactive git-receive-pack git-reflog git-
relink git-remote git-repack git-repo-config git-request-pull git-rerere git-
reset git-rev-list git-rev-parse git-revert git-rm git-send-email git-send-
pack git-sh-setup git-shell git-shortlog git-show git-show-branch git-show-
index git-show-ref git-stash git-status git-strip-space git-submodule git-svn
git-symbolic-ref git-tag git-tar-tree git-unpack-file git-unpack-objects git-
update-index git-update-ref git-update-server-info git-upload-archive git-
upload-pack git-var git-verify-pack git-verify-tag git-web--browse git-
whatchanged git-write-tree gitk

why git?

git advantages

crazy fast

works disconnected

allows many distribution models

as granular as you need

git encourages agile

workflow #1:
jumping in

~/.gitconfig

```
[user]
  name = Stuart Halloway
  email = stu@thinkrelevance.com
[alias]
  co = checkout
[apply]
  whitespace = nowarn
[core]
  excludesfile = /Users/stuart/.gitignore
[color]
  ui = auto
```

create a repos

```
mkdir sample  
cd sample  
git init
```

Initialized empty Git
repository in .git/

```
ls -G .git
```

```
HEAD  
config  
description  
hooks  
info  
objects  
refs
```

entire repos
is *local*

add some
content

```
git add README
```

(no output)

git status

```
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..."
#    to unstage)
#
# new file:   README
#
```

commit

```
git commit -m  
"initial commit"
```

```
Created initial commit  
   cfaeela: initial commit  
1 files changed,  
1 insertions(+),  
0 deletions(-)  
create mode 100644 README
```

have a look
around

git log

```
commit cfaee1a7f50c6cf4443fe5c0495ccaace670e2a8
Author: Stuart Halloway
       <stu@thinkrelevance.com>
Date:   Wed Mar 11 07:50:17 2009 -0400

    initial commit
```

commit
"numbers" are
shas

all git names are
shas!

```
git show cfaee1
```

```
commit cfaee1a7f50c6cf4443fe5c0495ccaace670e2a8
```

```
Author: Stuart Halloway
```

```
<stu@thinkrelevance.com>
```

```
Date:    Wed Mar 11 07:50:17 2009 -0400
```

```
initial commit
```

```
diff --git a/README b/README
```

```
new file mode 100644
```

```
index 0000000..af0649b
```

```
--- /dev/null
```

```
+++ b/README
```

```
@@ -0,0 +1 @@
```

```
+Sample project to demonstrate git workflows
```

commits point to
trees

```
git cat-file -p  
cfaeel
```

```
tree 1422b871e03802944801431721473b70d36ac017  
author Stuart Halloway  
    <stu@thinkrelevance.com> 1236772217 -0400  
committer Stuart Halloway  
    <stu@thinkrelevance.com> 1236772217 -0400  
  
initial commit
```

commits point to
parent commits

create another
commit

```
git add INSTALL
git commit
```

```
Created commit 34b5896:
  installation instructions
1 files changed,
1 insertions(+), 0
deletions(-)
create mode 100644 INSTALL
```

```
git cat-file -p  
HEAD
```

```
tree 108fb1051363bcef2c3d89cc426917396a8ede92  
parent cfae1a7f50c6cf4443fe5c0495ccaace670e2a8  
author Stuart Halloway  
    <stu@thinkrelevance.com> 1236774551 -0400  
committer Stuart Halloway  
    <stu@thinkrelevance.com> 1236774551 -0400  
  
installation instructions
```

HEAD is a local
name for tip of
current branch

trees point to
trees and *blobs*

```
git ls-tree 34b5896
```

```
100644 blob f4522df7cf4e299095d4491b5e58a12745230c14 INSTALL  
100644 blob af0649b4e1b3a1f684aa99d792c34c77182e30ea README
```

blobs hold
file contents

workflow #2:
shared remote,
branch for stories

clone the
remote repos

```
git clone some_url
```

```
Initialized empty Git repository in  
  /Users/stuart/tmp/sample/.git/  
remote: Counting objects: 6, done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 6 (delta 0), reused 0 (delta 0)  
Receiving objects: 100% (6/6), done.
```

```
git branch -a
```

- * master
- origin/HEAD
- origin/master

create a
local branch

```
git checkout -b  
Story23
```

```
Switched to a new branch  
"Story23"
```

work on the local
branch

```
git add Story23.java  
git commit -m 'story'
```

```
Created commit cc4aca4:  
    story 23 completed  
1 files changed,  
1 insertions(+),  
0 deletions(-)  
create mode 100644  
Story23.java
```

merge back
to master

```
git checkout master
```

```
Switched to branch "master"
```

```
git merge story23
```

```
Updating 34b5896..cc4aca4
```

```
Fast forward
```

```
Story23.java |      1 +
```

```
1 files changed ...
```

```
create mode 100644
```

```
Story23.java
```

meanwhile,
john doe
implements
story24

i push
story23 first

git push

```
Counting objects: 4, done.  
Compressing objects: 100% (2/2), done.  
Writing objects: 100% (3/3), 363 bytes, done.  
Total 3 (delta 0), reused 0 (delta 0)  
Unpacking objects: 100% (3/3), done.  
To file:///Users/stuart/gitrepos/sample  
    34b5896..cc4aca4  master -> master
```

john doe
must pull, then
push

git pull

```
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
From file:///Users/stuart/gitrepos/sample
   34b5896..cc4aca4  master    -> origin/master
Merge made by recursive.
 Story23.java |      1 +
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 Story23.java
```

john has a
merge commit

git log

```
commit 3fb73c1b9e09633512d9be25a24280c092e28400
Merge: b3eab67... cc4aca4...
Author: John Doe <jdoue@thinkrelevance.com>
Date:   Wed Mar 11 08:34:16 2009 -0700
```

```
    Merge branch 'master' of
    file:///Users/stuart/gitrepos/sample
```

```
commit b3eab67db5b8f7ef226681d702eac62ef7d22eaa
Author: John Doe <jdoue@thinkrelevance.com>
Date:   Wed Mar 11 08:33:22 2009 -0700
```

```
    completed story 24
```

john pushes
story24 and
merge

git log

```
commit 3fb73c1b9e09633512d9be25a24280c092e28400
Merge: b3eab67... cc4aca4...
Author: John Doe <jdoue@thinkrelevance.com>
Date:   Wed Mar 11 08:34:16 2009 -0700
```

```
    Merge branch 'master' of
    file:///Users/stuart/gitrepos/sample
```

```
commit b3eab67db5b8f7ef226681d702eac62ef7d22eaa
Author: John Doe <jdoue@thinkrelevance.com>
Date:   Wed Mar 11 08:33:22 2009 -0700
```

```
    completed story 24
```

where are we?

git as filesystem

git	traditional filesystem
blob	file
tree	directory
commit	<i>total history!!</i>

content addressing

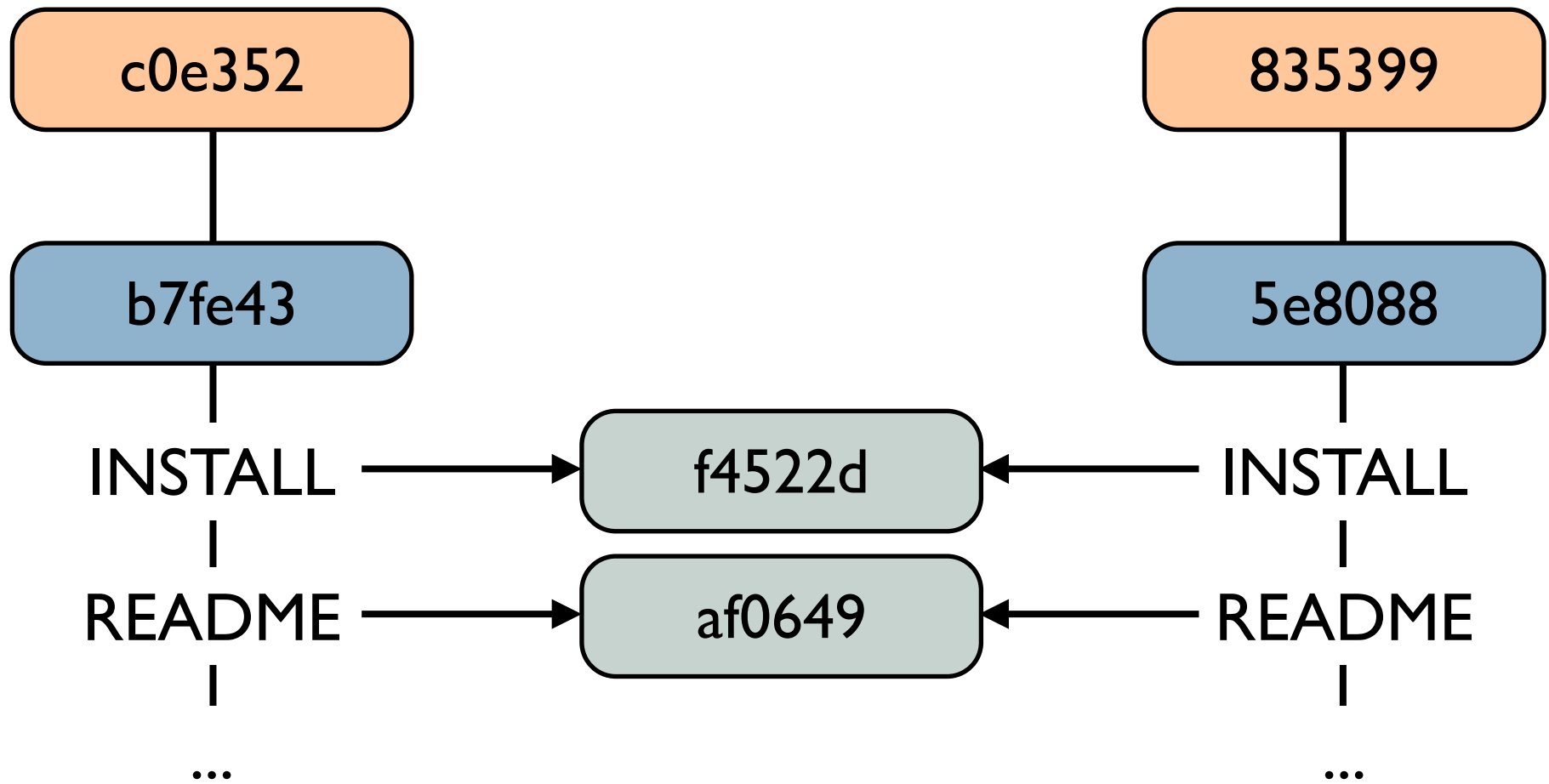
blobs are immutable

trees are immutable

commits are immutable

blobs and trees *share structure*

sharing structure



almost a free lunch

shared structure saves memory

operations on pointers are crazy fast

everyone has the full repos

work disconnected

no single source of truth

content addressing never lies

workflow #3:
multiple remotes,
conflicts

add one or
more remote
branches

```
git remote add  
ringill some_url
```

(ringill is now alias
for some_url)

create trial branch
for the merge

```
git checkout -b  
try-ringill
```

Switched to a new branch
"try-ringill"

fetch from the
new remote

```
git fetch ringill
```

```
remote: Counting objects: 5, done.  
remote: Compressing objects: 100% (3/3), done.  
remote: Total 3 (delta 1), reused 0 (delta 0)  
Unpacking objects: 100% (3/3), done.  
From file:///Users/stuart/tmp/remote-2  
* [new branch]      Story24      -> ringill/Story24  
* [new branch]      master       -> ringill/master
```

merge into
the trial branch

```
git merge  
ringill/master
```

Auto-merged README

```
CONFLICT (content): Merge conflict in README  
Automatic merge failed; fix conflicts and  
then commit the result.
```

review and
fix conflicts

conflicts

Sample project to demonstrate git workflows

```
<<<<<<< HEAD:README
```

```
Real Contributors:
```

```
  Stuart Halloway
```

```
Fictional Contributors:
```

```
  John Doe
```

```
  Takeshi Kovacs
```

```
=====
```

```
See also http://github.com/relevance/git-control
```

```
>>>>>>> ringill/master:README
```

add and commit
unmerged files

git status

```
README: needs merge
# On branch try-ringill
# Changed but not updated:
#   (use "git add <file>..."
#   to update what will be committed)
#
# unmerged:   README
#
```

```
git commit -a
```

(opens your default editor to edit commit msg)

```
Created commit fec3fd5:  
Merge commit 'ringill/master' into try-ringill
```

it worked!
merge into master

```
git checkout master
git merge try-ringill
```

```
Updating 2d50db6..fec3fd5
```

```
Fast forward
```

```
README | 2 +-  
|
```

```
1 files changed, ...
```

workflow #4: revisionist history

lots of work
on story 26

```
git log
--pretty=oneline
```

```
c10bf0b7c823cd39f4e286ba52f2a8d6d63b2100
```

```
revised part 3 completed part 4 and 5 story complete!
```

```
0c2754e7a34fdecabc9ad4021ee69c6c211f86538
```

```
progress on part 3
```

```
76478e1a5ef090014b4578cb0b3ed177d6a692de
```

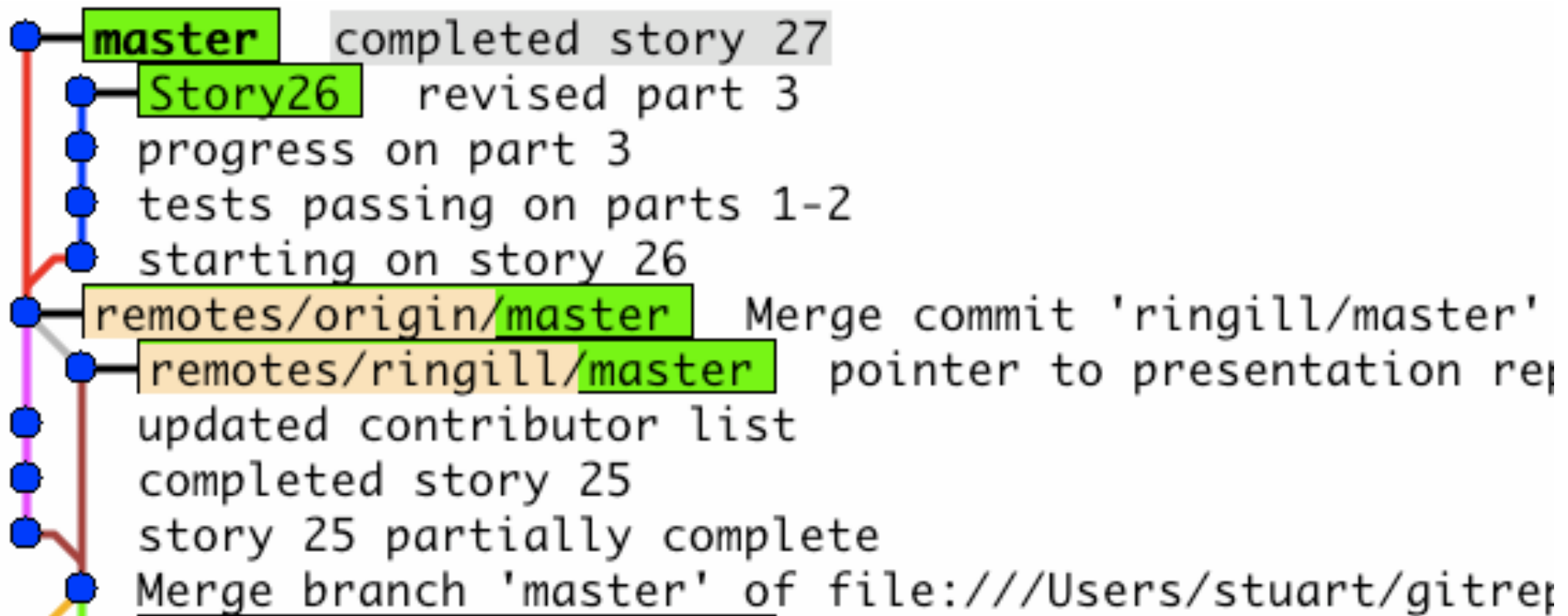
```
tests passing on parts 1-2 not started on parts 3-5
```

```
eded85b59cf71561f3c82e1bfc05c0ff7c64fc83
```

```
starting on story 26
```

meanwhile, ras
completes
story27

gitk visualization

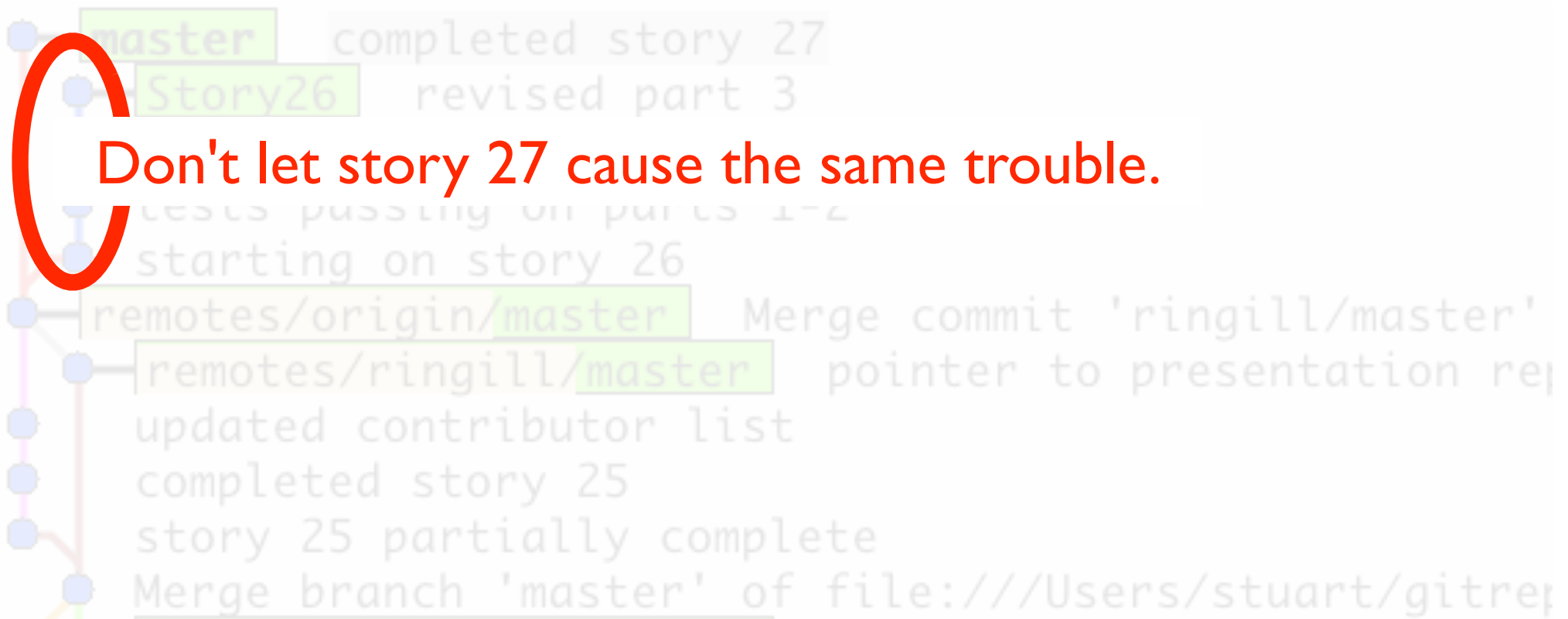


gitk visualization



Yuck. Noise from story 25 polluting main history

gitk visualization



rebase
master into
Story26

git rebase master

First, rewinding head to replay your work on top of it...

HEAD is now at d9fb455 completed story 27

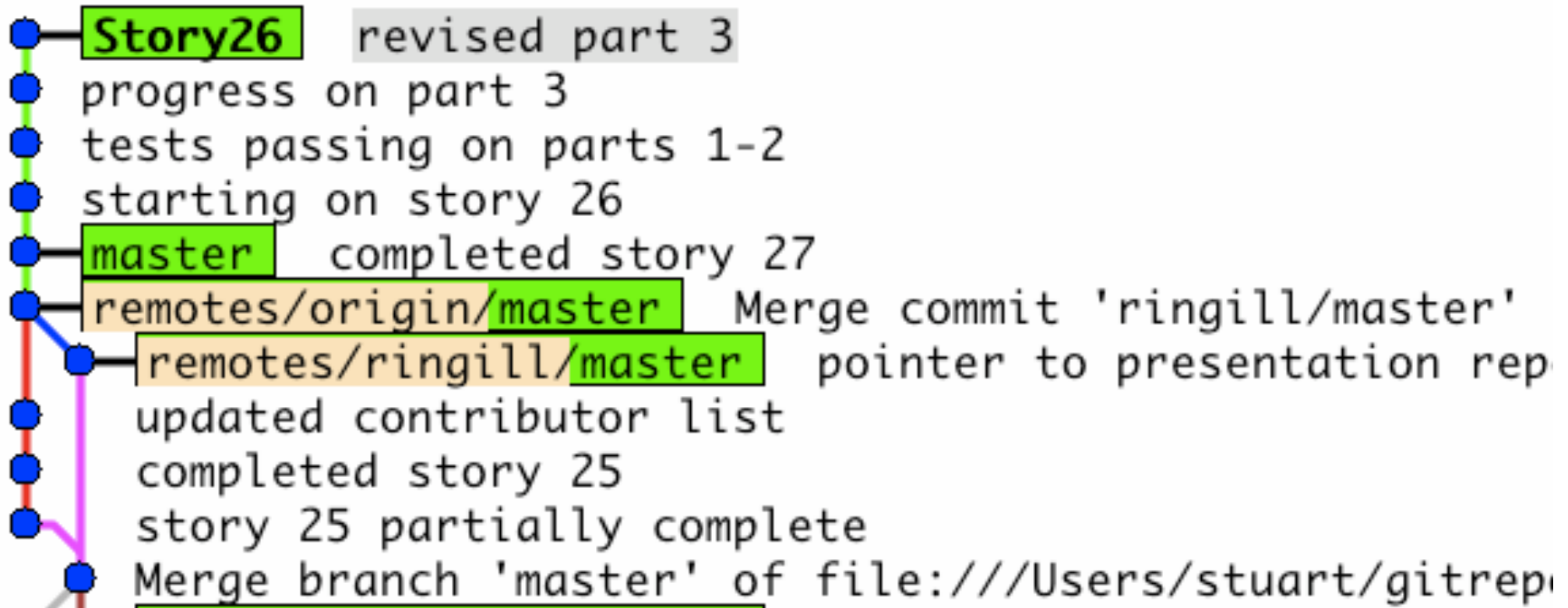
Applying starting on story 26

Applying tests passing on parts 1-2

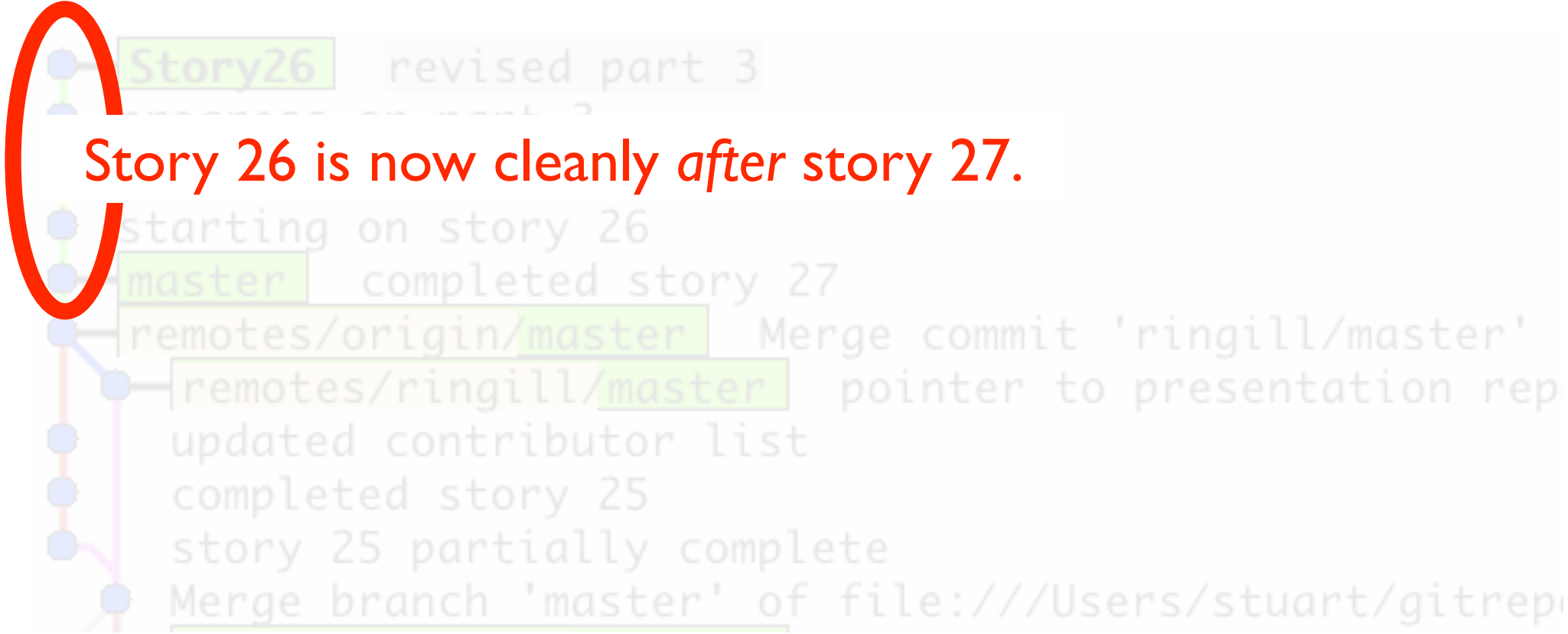
Applying progress on part 3

Applying revised part 3

history, changed



history, changed



Story 26 is now cleanly *after* story 27.

never change
public history

squash story26
commits into a
single commit
on master

```
git merge --squash  
Story26
```

```
Updating d9fb455..effe391  
Fast forward  
Squash commit -- not updating HEAD  
 story26.c |      9 ++++++++  
 1 files changed,  
 9 insertions(+),  
 0 deletions(-)  
create mode 100644 story26.c
```

```
git commit -m  
"completed story 26"
```

```
Created commit dba0ccd: completed story 26  
1 files changed,  
9 insertions(+),  
0 deletions(-)  
create mode 100644 story26.c
```

```
git branch -D Story26
```

```
Deleted branch Story26.
```

simplified history

dba0ccd1914d967f082ef4dcae6ec48d39b4b4ab

completed story 26

d9fb4553f6cf9d0a5f7592e85b555ed7abe994d8

completed story 27

fec3fd52f3480f238b5427a409a8043f694572c1

Merge commit 'ringill/master'
into try-ringill

the politics of git

forking: good

testing: more important than ever

code: naturally open

leadership: based on merit



Source Commits Info Wiki (1) **Network (5)** Admin

Graph Members Feed

relevance / tarantula

[edit](#) [pull request](#) [unwatch](#) [download](#)



Description: a big hairy fuzzy spider that crawls your site, wreaking havoc [edit](#)

Homepage: <http://opensource.thinkrelevance.com/wiki/tarantula> [edit](#)

Public Clone URL: <git://github.com/relevance/tarantula.git>

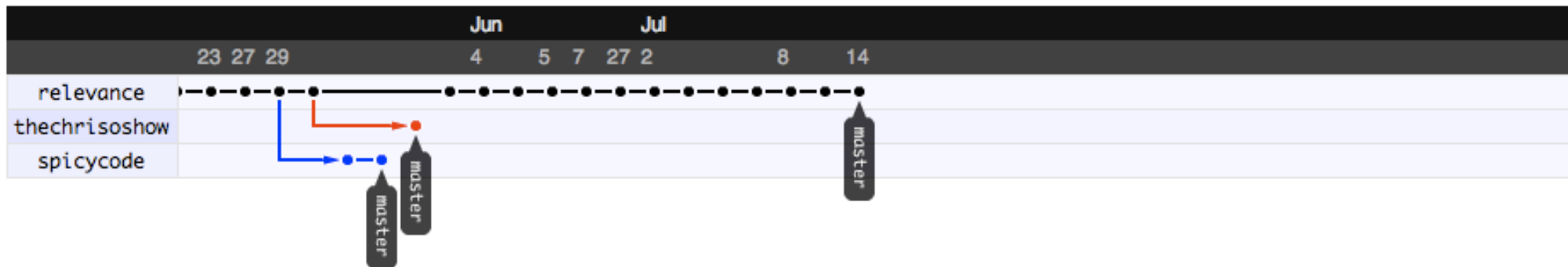
Your Clone URL: <git@github.com:relevance/tarantula.git>

Search Repo:

The tarantula network graph

All branches in the network using **relevance/tarantula** as the reference point. [Read our blog post about how it works.](#)

[Show Help](#)



resources

Chacon, Scott. <http://onrails.org/articles/2008/06/01/railsconf-2008-git-talk-by-scot-chacon-video>. Git presentation at RailsConf

Chacon, Scott. <https://peepcode.com/products/git-internals-pdf>. Git internals.

Grosenbach, Geoffrey. <http://peepcode.com/products/git>. Git peepcode screencasts.

Halloway, Stuart. <http://github.com/relevance/git-control>. This presentation and accompanying samples.

Torvalds, Hamano, et al. <http://git.or.cz/>. Git