Introduction to source control with git

Sam Procter

Department of Computing and Information Sciences
Kansas State University

Tuesday, September 16, 2014

Before we begin

1. I'm not a git expert
   - But I use git *every* day at my job
2. You can learn how to use git on your own
   - I'll show you a basic introduction and give you an *understanding* of git
3. Follow along at:
   http://people.cis.ksu.edu/~samprocter/git-intro.pdf

Outline

1. Source Control Management
   - A history of Source Control Tools
2. git
   - Git Concepts
   - Getting and Making Changes
   - Getting a Repository
   - Software Engineering with Github
3. Be Brave!
   - Learn by playing

Why use source control?

How do you share projects now?
What happens if. . .

- You delete some code but realize later that you need it?
- You do some work on a lab machine, and then want to work on your pc at home?
- Your laptop dies?
- Two people change the same file?

Solution: Source Control!

A super-brief history of source control

Gen.	Networking	Operations	Concurrency	Examples
1	None	One file at a time	Locks	SCCS
2	Centralized	Multi-file	Merge before commit	CVS, SVN
3	Distributed	Changesets	Commit before merge	git, hg

Table: Source control through the years, adapted from [1]

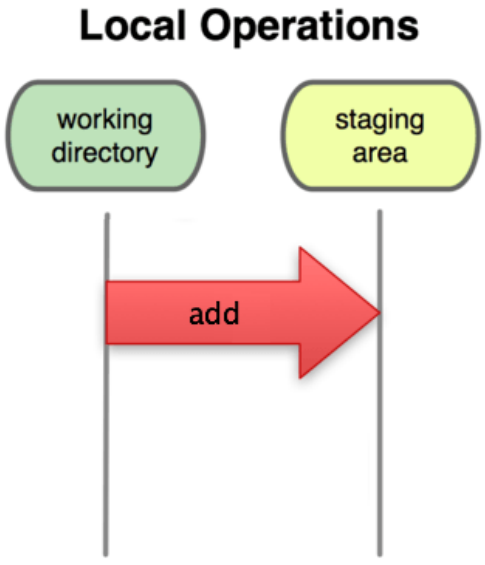
1. SCCS: 1972, File locks, no sharing at all
2. CVS, SVN: 1986 (CVS), deltas, merging
3. git, Mercurial: 2005, "file system snapshots" distributed

Where did git come from?

"Git began with a bit of creative destruction and fiery controversy"[2]

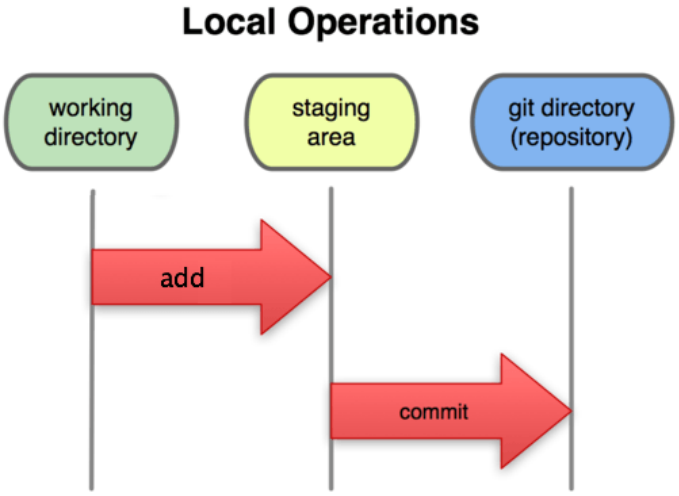
Git was created by Linus Torvalds (and other Linux developers) with aims of:

- Speed
- Simplicity
- Distributivity



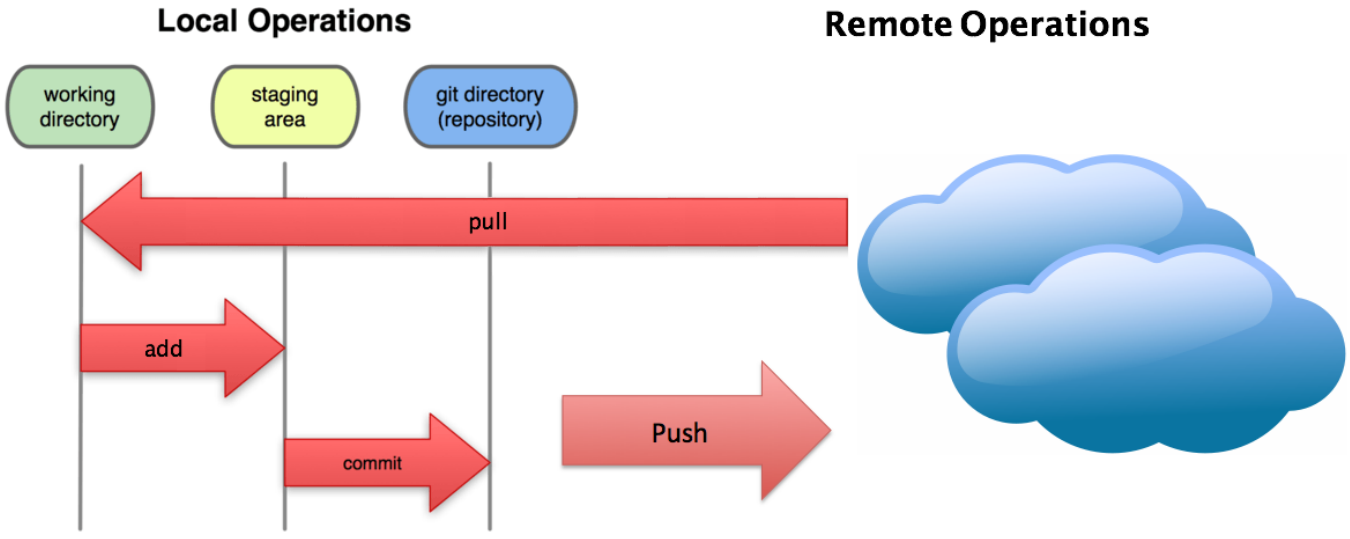
- Working Directory: Your modified files
- Staging Area: Files to be shipped with the next commit
- Git Directory: The official record
- The cloud: github, cis.ksu.edu/~yourname, etc.

Git Concepts



- Working Directory: Your modified files
- Staging Area: Files to be shipped with the next commit
- Git Directory: The official record
- The cloud: github, cis.ksu.edu/~yourname, etc.

Git Concepts



- Working Directory: Your modified files
- Staging Area: Files to be shipped with the next commit
- Git Directory: The official record
- The cloud: github, cis.ksu.edu/~yourname, etc.

Sharing changes with git commit & push

Situation: You've created two files—a.txt, b.txt—and you want to share both of them.

Strategy: Use `git add` on a.txt and b.txt.

1. `$ git add a.txt b.txt`
2. `$ git commit -m 'Two new files!'`
3. `$ git push`

Sharing changes quickly

Situation: You've changed files, but they're already tracked by git.

Strategy: Use `git commit` with the `-a` flag to add everything automatically

1. `$ git commit -a -m 'Changed things'`
2. `$ git push`

Getting changes

Situation: You made changes at school, now you want to get those at home.

Strategy: Use `git pull` to get the latest version of your files

1. `$ git pull`

Git Init – You have files

Situation: You've created some files, but never stored them in git.

Strategy: Use `git init` to create a repository, and `git push` to share it

1. `$ git init`
2. `$ git add .`
3. `$ git commit -m 'First commit'`
4. `$ git remote add origin <remote repo URL>`
5. `$ git push origin master`

(Adapted from [3])

Git Clone – You need files

Situation: Files you want are on github, and you've never downloaded them before.

Strategy: Use `git clone` to get the files for the first time

1. `$ git clone <remote repo URL>`

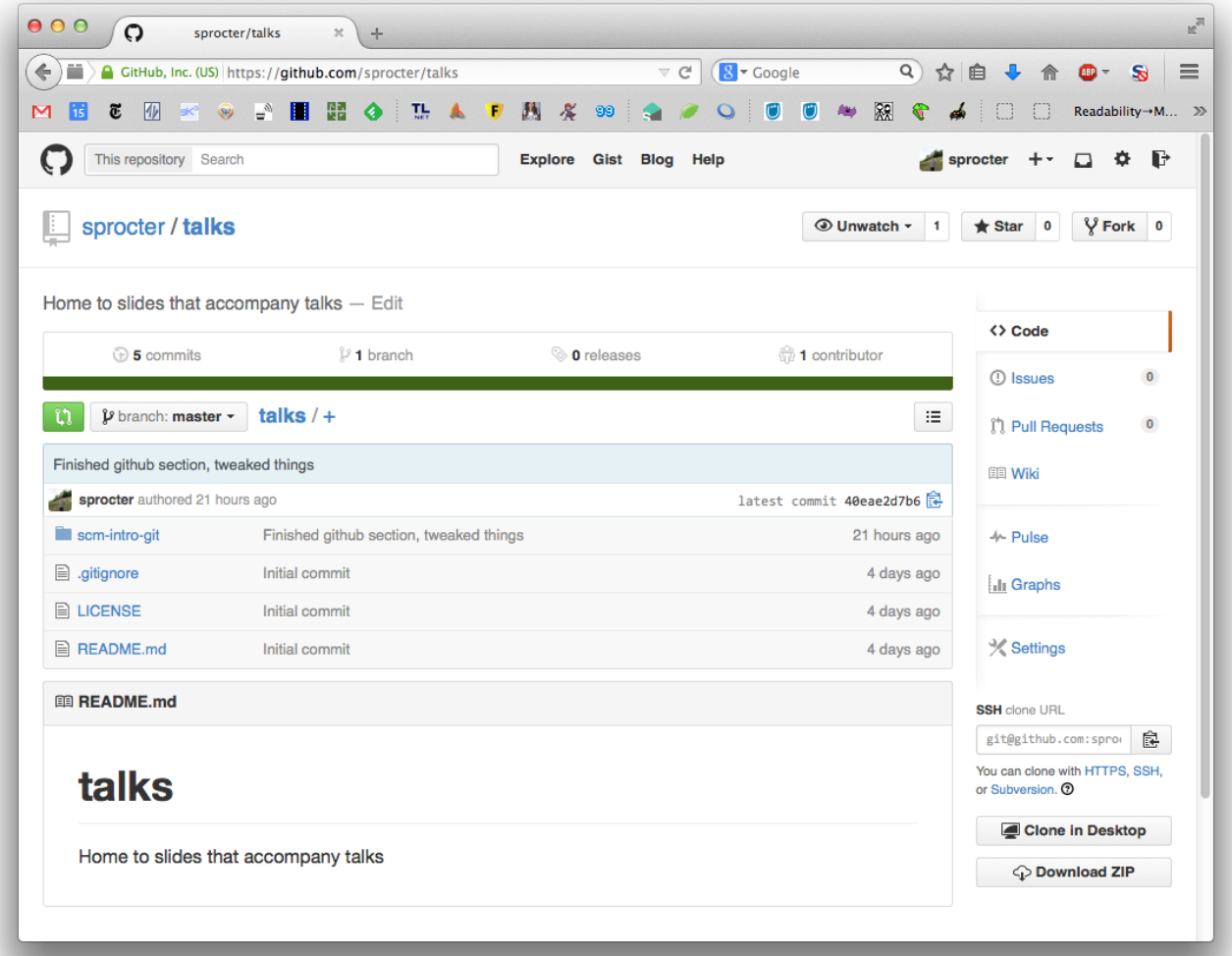
What is github?

Github is a free* git host.

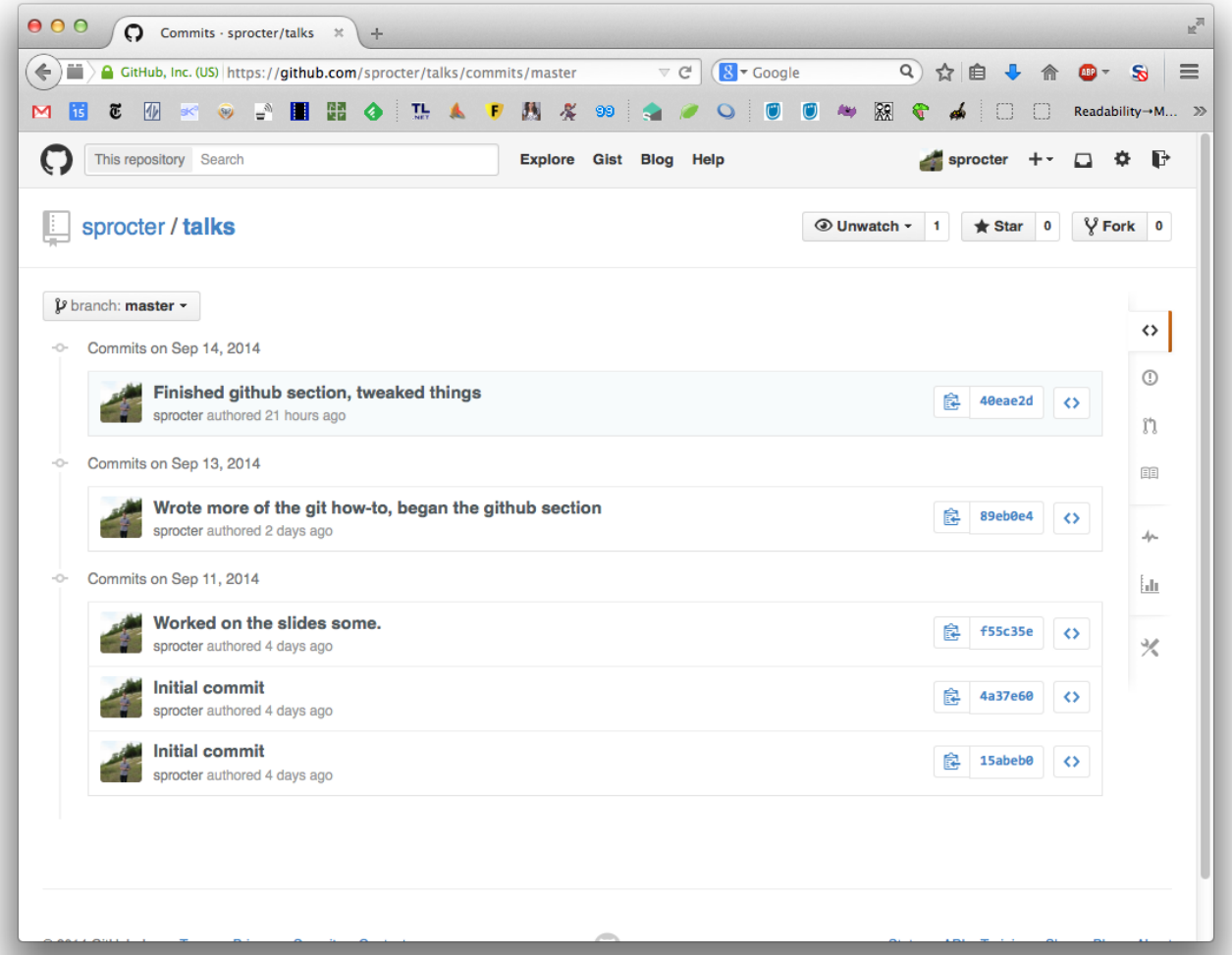
* Free means:

- Infinite (∞!) publically-accessible repositories
- Five free private repositories for students

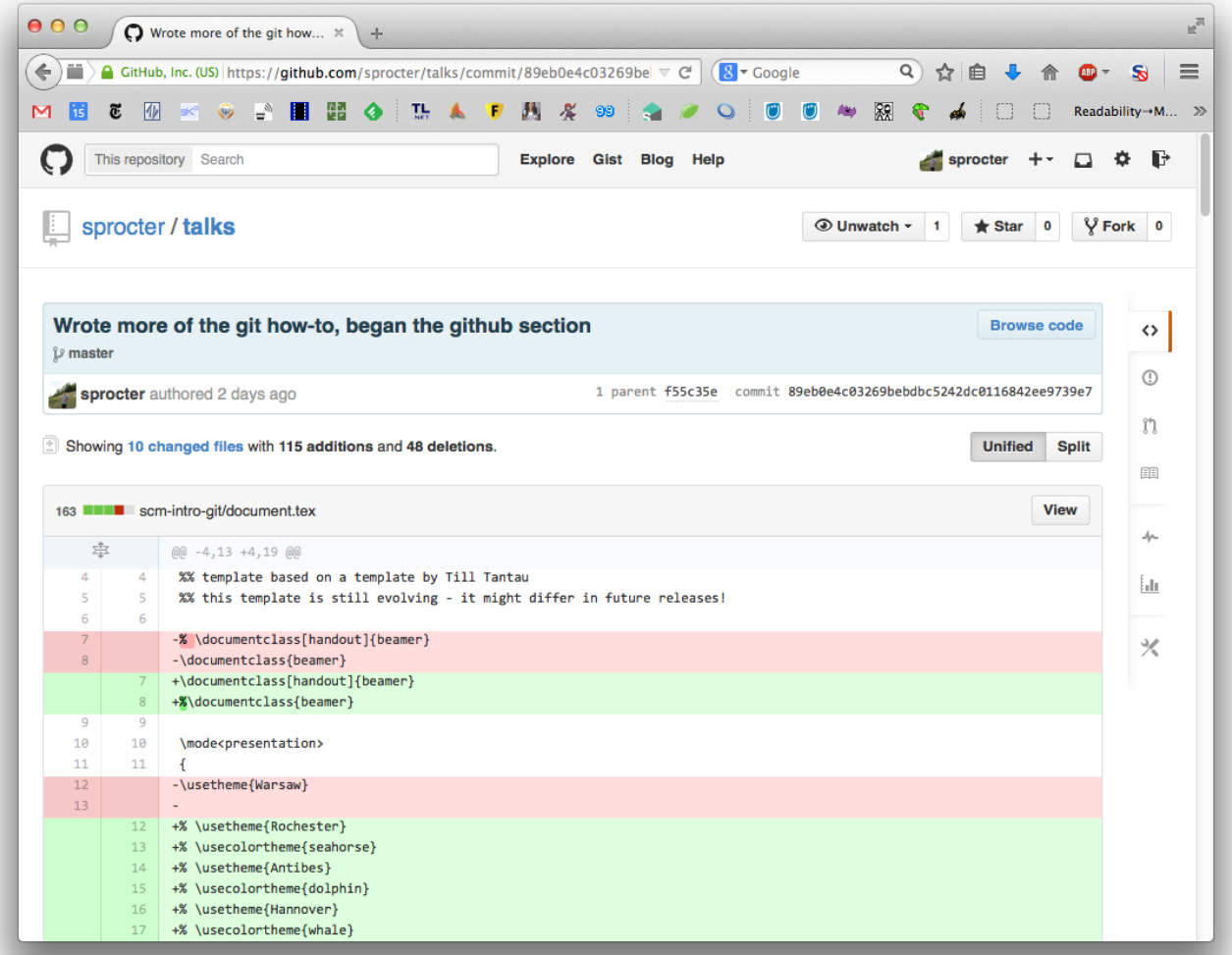
Using github



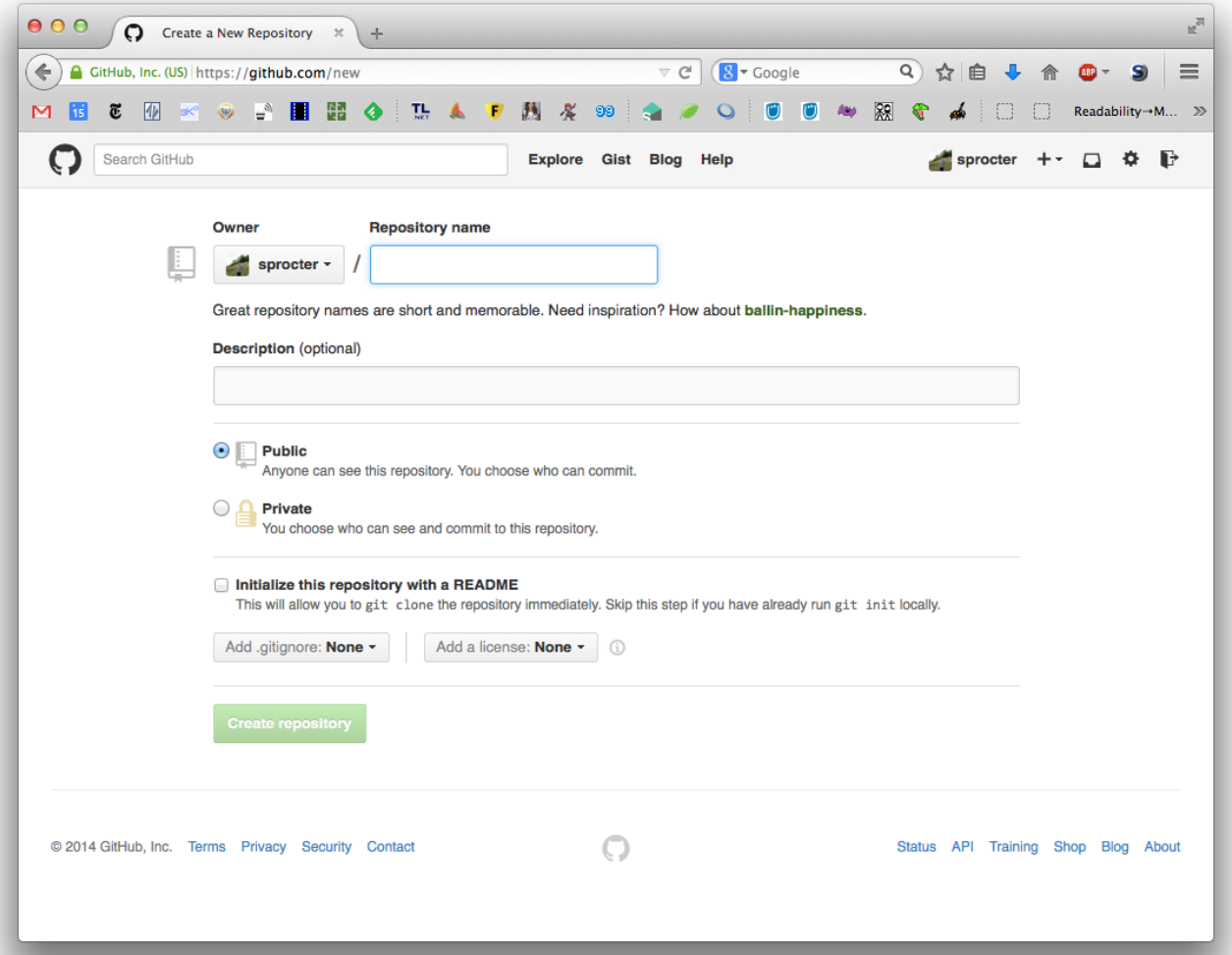
Viewing the list of commits



Viewing the details of a commit



Creating a new repository with github



Be Brave!

- git is *tremendously* googlable.
- Breaking things (especially after a commit / push) is very difficult
- Play with it!

No seriously, try it right now with either:

- https://try.github.io
- Your own github account

References I


Eric Sink.
Version control by example.
Pyrenean Gold Press, 2011.


Scott Chacon.
Pro Git.
Apress, 2009.


Github, Inc.
`https://help.github.com`