

Source Control 101

with GitHub

Max Annear-Henderson

Luke Ryan <http://lukeryan.net.nz>



Beautiful accounting software

Imagine...

Imagine you worked on a team with a product that had 500,000 lines of code over thousands of files with dozens of developers all around the world?

How would you work together as a team? How would you share files?

Why source control?

Teamwork

- “I’m working on file X, nobody else touch it”
- Multiple people can work on the same files, then you can merge the changes together at integration time.
- Full history of who changed what, when. No lost code.

Why source control?

Backups

Repository of code - one source of truth

Backed up

“I lost my laptop...”

“I deleted the shared directory...”

“The production server died...”

Why source control?

Change management

- History of revisions over time - who changed what and when
- When was a defect introduced?
- What code is running on environment x?
- The only way to maintain changes to multiple different versions at the same time
- No need to comment out code!

Why source control?

Rapid innovation

With your source code backed up, you can do crazy things

“what happens if I delete all of these dependencies then re-write this class...”



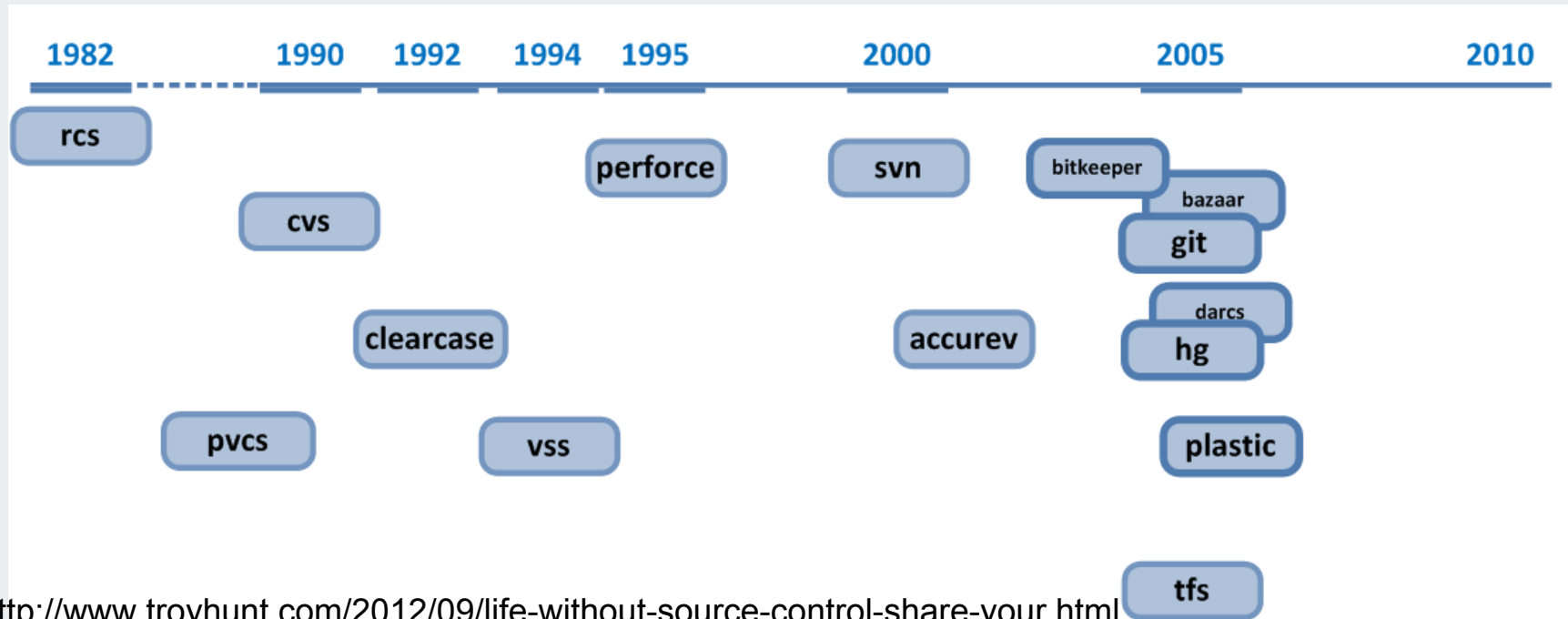
Why source control?

Automate all the things

Once you have a centralised repository of your code you can automate builds, testing, deployment from it.

Triggered by changes.

There are lots of source control systems



Git

- git: distributed source control
- you have a local copy of the repository
- can push to remote server repository

Written by some famous nerd...



in 3 weeks.



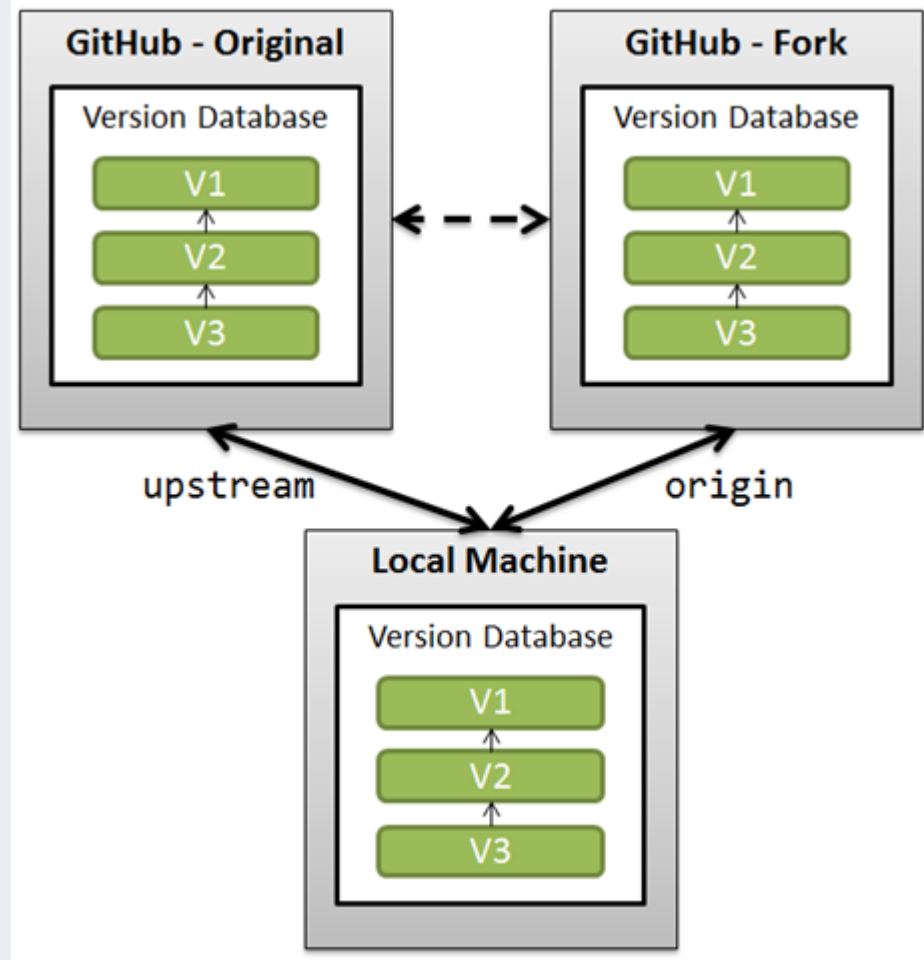
GitHub



- Sweet online version of git
- Makes social coding easy (pull requests)
- Demo

Forks

- Isolated server copy of the repository
- Useful if you don't have permission to push to the remote repository
- Demo



Git Clients

- Git Bash (Command line)
 - GitHub for windows/mac
 - GitExtensions
 - Atlassian SourceTree
-
- Today we are using Git Bash

Your turn - Fork

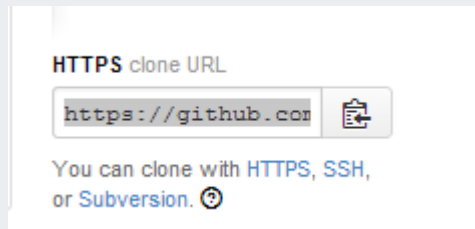
Create a github account at <http://github.com>

Go to <https://github.com/lukeryanxero/summer-of-tech>

Click 'Fork Repository'

Your turn - Clone

- Clone it locally
 - Get HTTPS clone URL from your github fork
 - Open Git command line
 - Type git clone [URL]

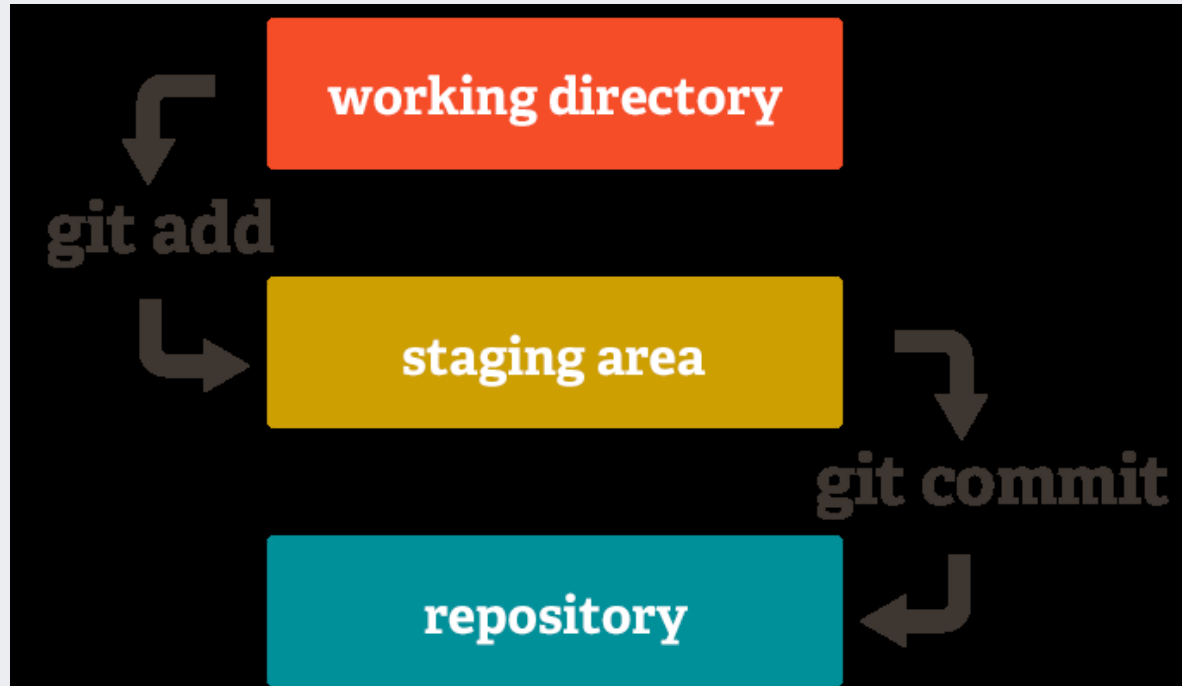


```
max.henderson@ /c/dev/sot
$ git clone https://github.com/max-xero/summer-of-tech.git
Cloning into 'summer-of-tech'...
remote: Counting objects: 6, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 6 (delta 0)
Unpacking objects: 100% (6/6), done.
```

Commit

- Has a unique hash
- Set of changes to many files
- Represents a piece of work from one person
- Has a comment - can include issue numbers

<http://bit.ly/1k43n47>



Your turn

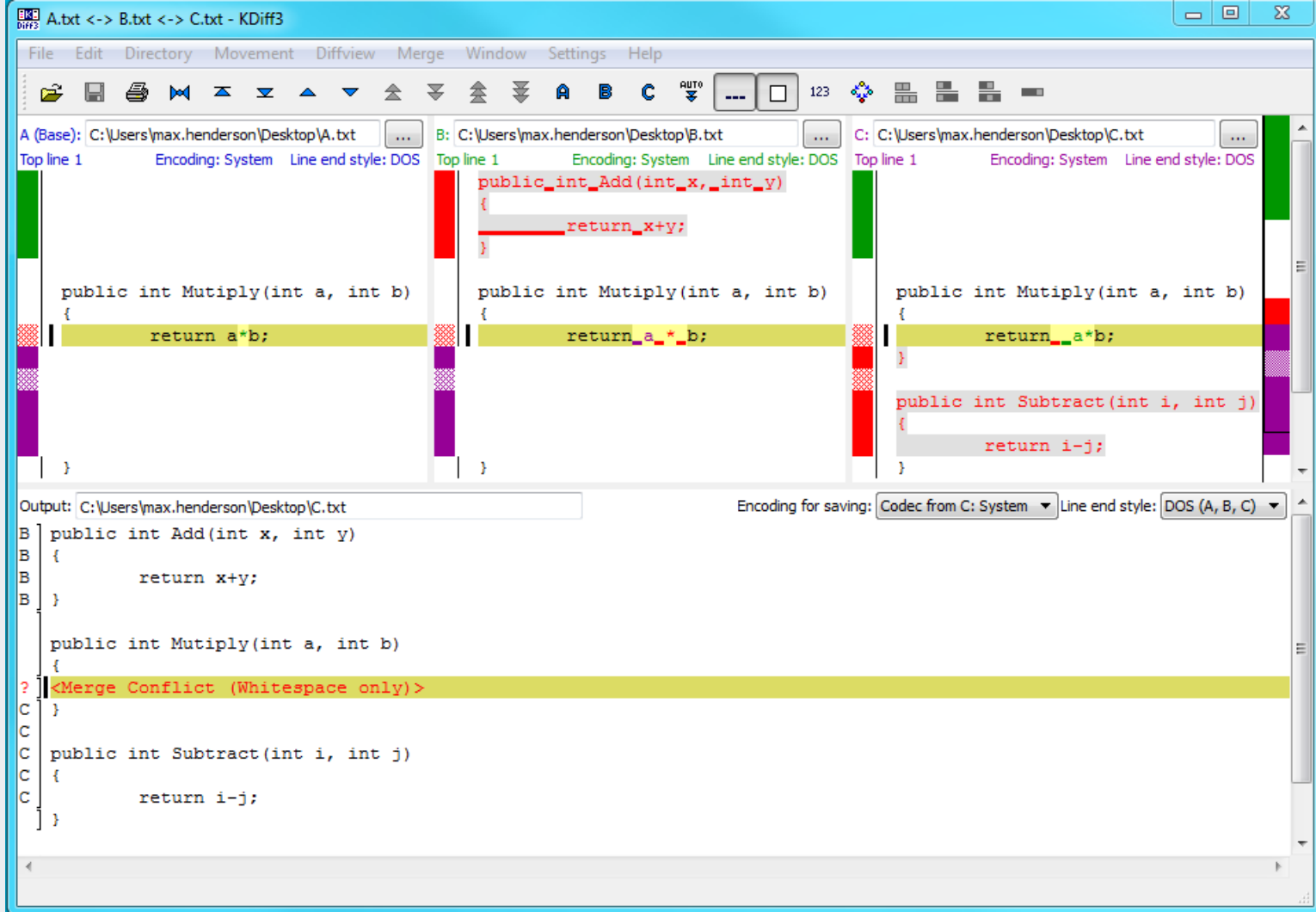
- Change the readme.MD file from the repo you cloned earlier
- `git status`
- `git add *`
- `git status`
- `git commit -m "Some awesome message"`

Push

Sends your changes (commits) back to the server (your fork)

- `git push origin`

Merging



Pull Requests

- Request to pull changes from our Fork to the master Repo.
- Internally we use this for code reviews

<https://github.com/lukeryanxero/summer-of-tech/pull/1>

Resources

<https://try.github.io>

<http://gitimmersion.com/>

<http://git-scm.com/book/en/Getting-Started-Git-Basics>

<https://www.youtube.com/watch?v=1ffBJ4sVUb4> 1:40:00

<http://gitref.org/index.html>

Branching

- Why
- What it is
- Branching strategies
- Change control and software versioning
- <http://nvie.com/posts/a-successful-git-branching-model/>