STAT 209
Interlude: Version Control

September 19, 2019

Colin Reimer Dawson

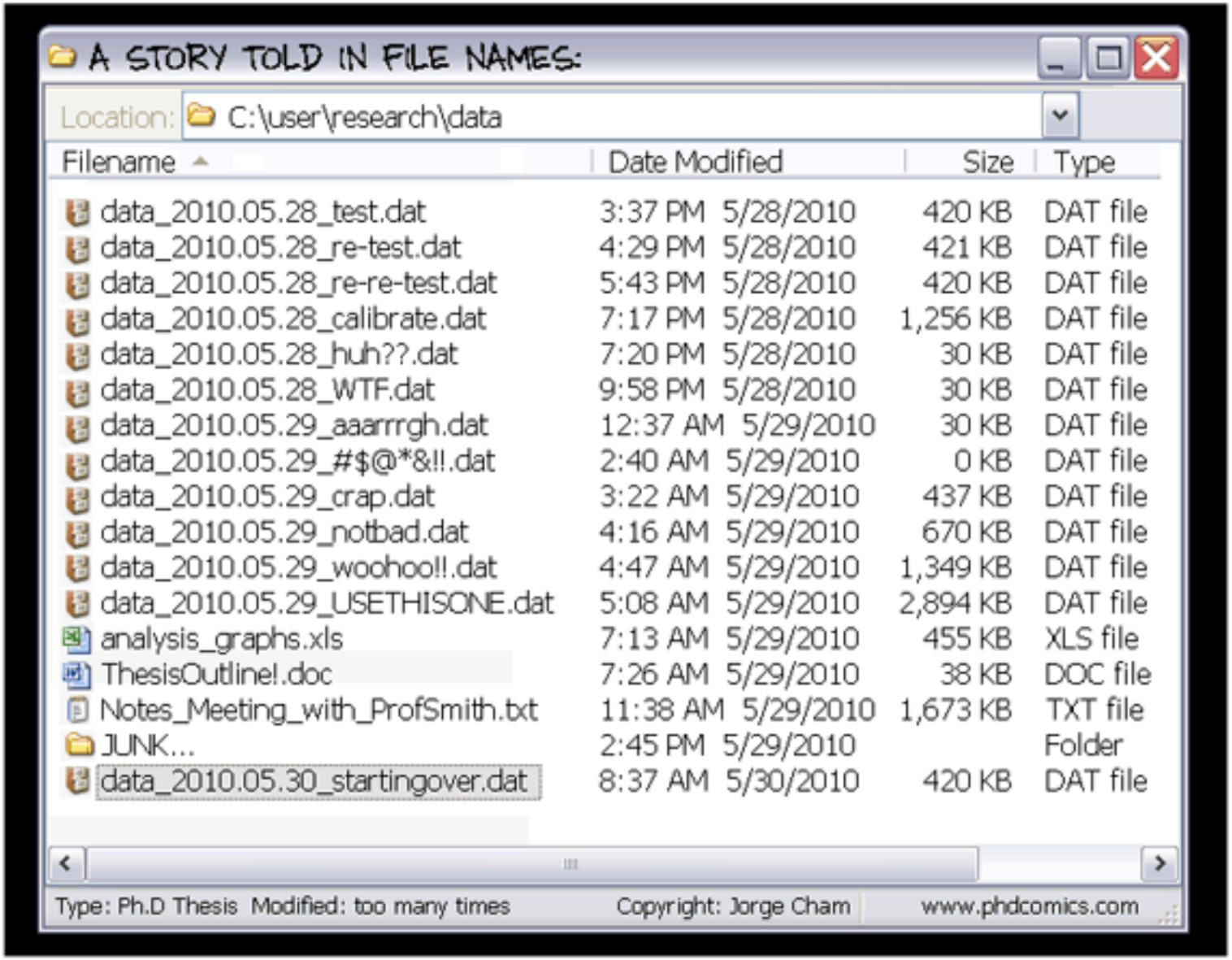
Happy International Talk like a Pirate Day!

What is Version Control?

A version control system...

- records the history of changes to your code
- facilitates collaboration by propagating changes among users
- enables reversion to an earlier project state

Why Version Control?



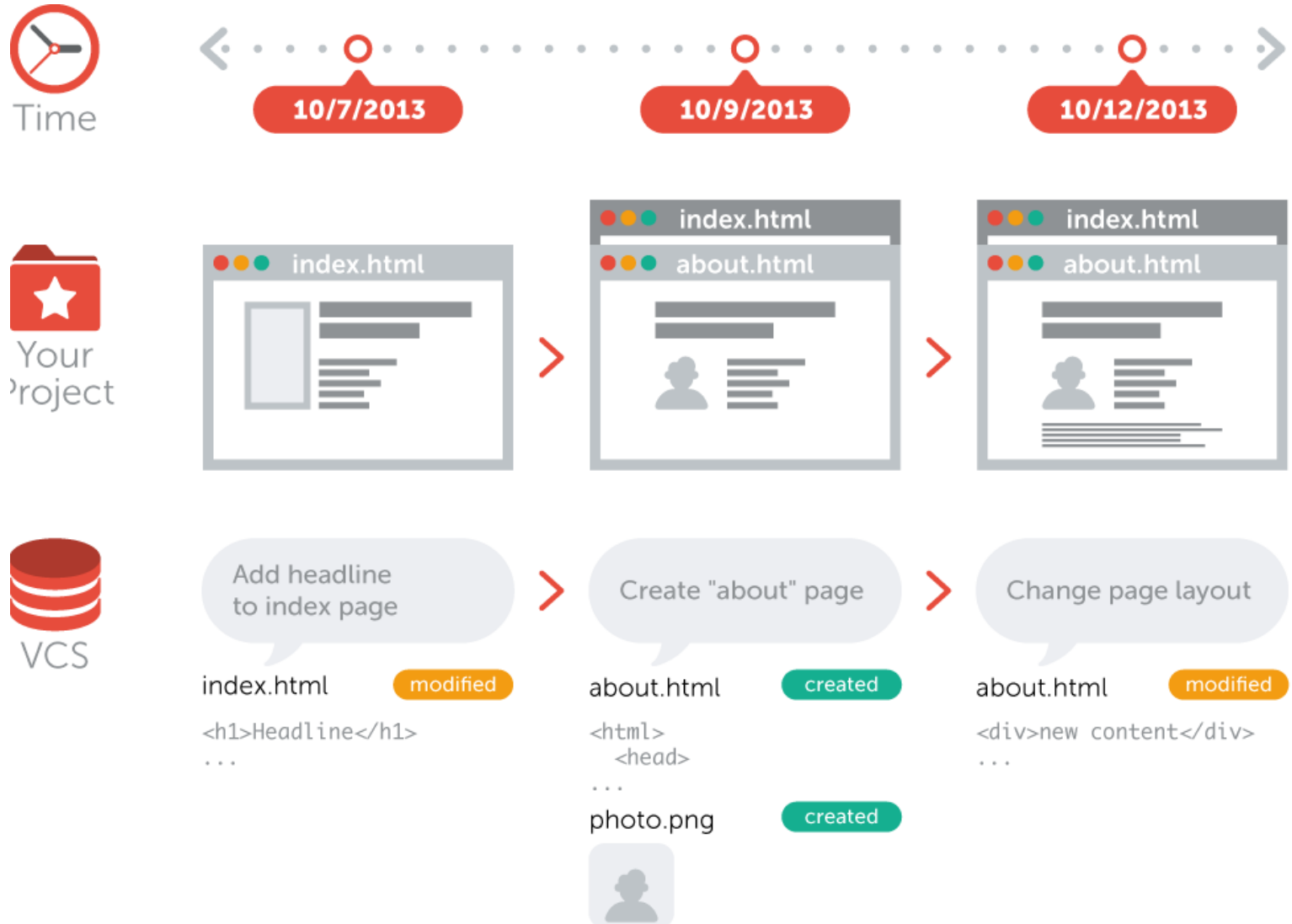
Version Control Systems

- Concurrent Versions Systems (CVS) – since 1986
- Subversion (`svn`) – since 2000
- Git – since 2005

Version Control Systems

- Concurrent Versions Systems (CVS) – since 1986
- Subversion (`svn`) – since 2000
- Git – since 2005
- Various others

How does it work?



Key Concept: Snapshots

- A record of what the files in your project look like at any given time
- You decide when to take a snapshot (called "committing" or "making a commit")
- The version control system allows you to revisit earlier snapshots

Key Concept: Commit

- As a verb, to create a new snapshot
  - "I committed my code"
- As a noun, the update from one snapshot to the next
  - "I made a commit"

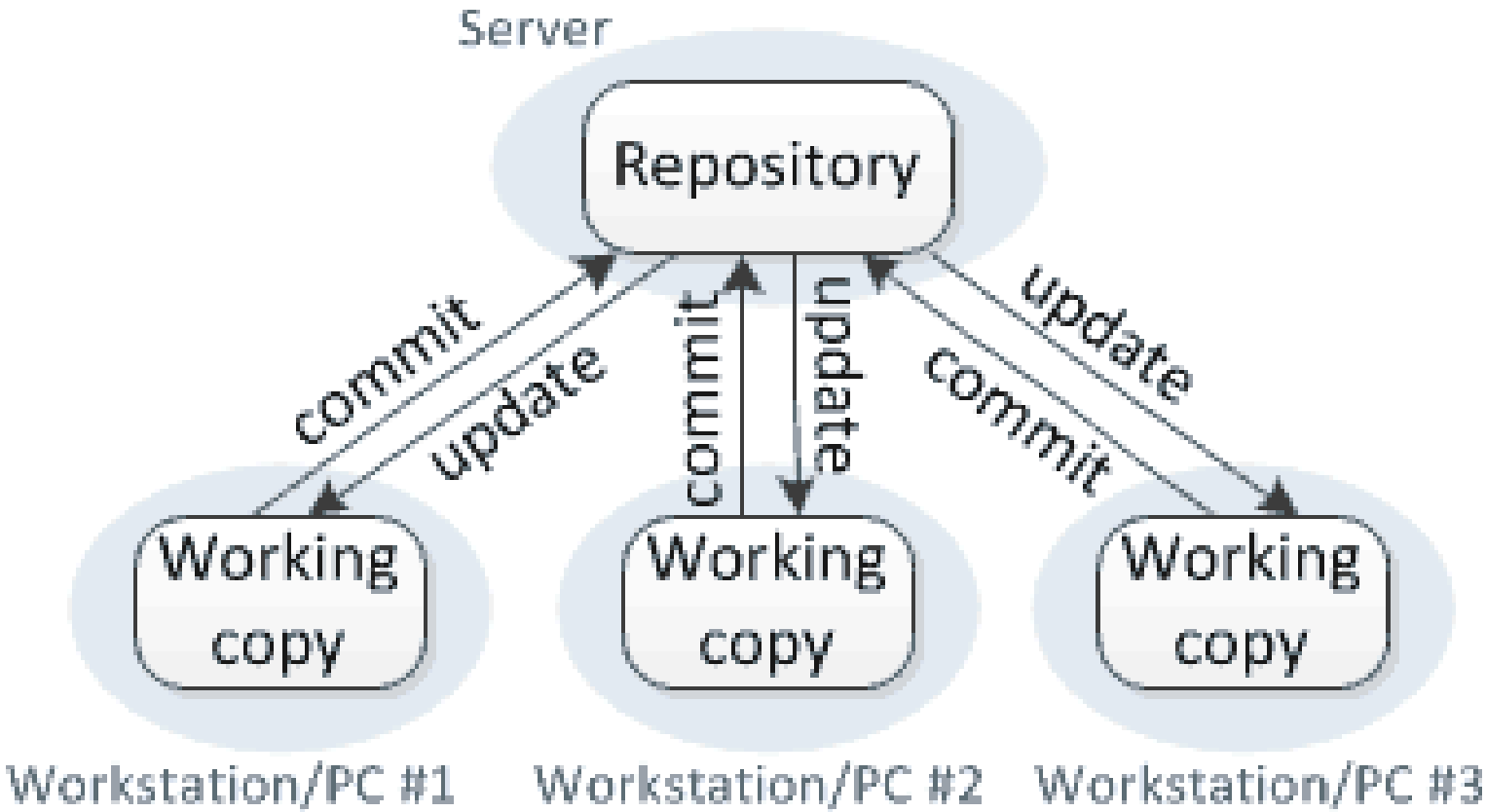
What's in a Commit?

1. A reference to the previous commit ("parent commit")
2. A record of changes since the last commit
3. A unique "hash code" identifier (a long string of letters and numbers like `2ff78d5ebaf48f43f7de26d1bcae52714fa23549`)

Key Concept: Repository

- A collection of files in a project, along with a version history of those files
    - Consists of all commits
- "repo" for short

Centralized version control

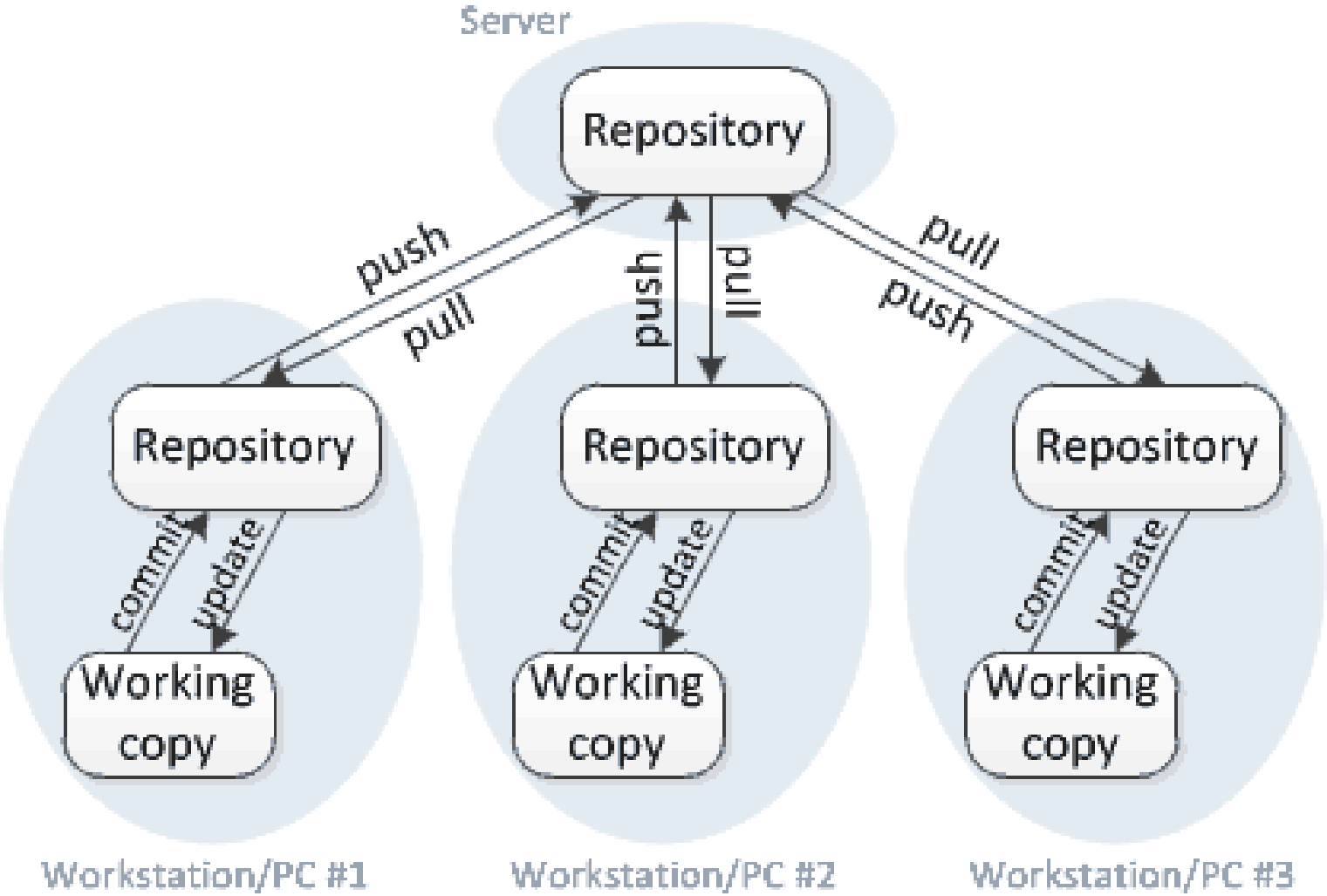


Subversion and CVS are like this

Git is Distributed

- In a distributed model, each user possesses a full (**local**) repository
- Changes can be propagated or **pushed** from a local repo to a **remote** repo on a server ("in the cloud")
- Changes can be fetched or **pulled** from the remote repo to the local repo
- GitHub is one place where git repos can be hosted remotely

Distributed version control



Git is like this

The Main Git Verbs

`clone`	Copy a remote repo to your computer
`pull`	"Download" changes from the remote repo to your local repo
`add`	Register changes to some files to be committed at the next commit
`commit`	Take a snapshot of your working directory and register the state in your local repo
`push`	"Upload" new commits from your local repo to the remote repo

Git in RStudio

- RStudio is integrated with Git and provides a graphical interface for it
- You can use the command line if you prefer, by opening a Shell in RStudio, but I will guide you through the graphical approach
- Interactive tutorials for the standard CLI at
  - `try.github.io`
  - DataCamp
  - Numerous other places