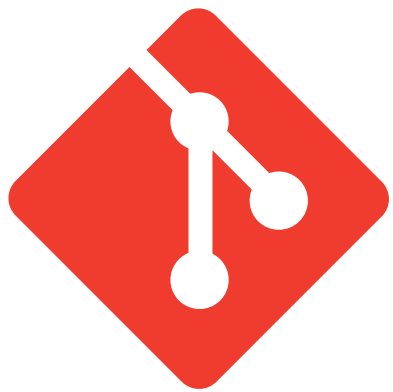
PURDUE
UNIVERSITY

Version Control Workshop: Git and GitHub



Cyrus Vandrevala [1]
Nicolás Guarín-Zapata[2]
[1] Physics Department
[2] Civil Engineering Department
October 30-31, 2014

Overview

1. Introduction to Version Control
2. Workflow in Computational Science
3. Learning Git
   - Setting Up Git On Your Machine
   - Basic Git Cycle
   - Git Branches
   - Git Delete Commands
4. Git and GitHub
5. Collaborated Summary

What is Version Control?

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

Why is Version Control Important?

1. Keep Track of Code History
2. Concurrent Teamwork
3. Coordinate Coding Environments
4. Due Diligence Checks
5. Share Code

Everybody Should Use Version Control!

What Options Are Available?

Option #1: Client-Server Version Control Systems

Advantages

1. A Single Admin Keeps Track of the Project
2. There is a Single Master Version of the Code
3. It is Relatively Easy to Learn

Disadvantages

1. There Is Only One Admin/Server
2. You Need a Network Connection to Work
3. Operations Can Be Slow

Examples include Concurrent Versions System (CVS) and Subversion (SVN).

What Options Are Available?

Option #2: Distributed Version Control Systems

Advantages

1. You Don't Need a Network Connection
2. Multiple Coding Environments
3. It Encourages Collaboration and Modularity

Disadvantages

1. Can Be Difficult to Learn
2. Teams Need to Talk About Conventions
3. It is Really Easy To Create Unorganized Code

Examples include Git/GitHub and Bazaar.

Why Git and GitHub?

1. It Keeps Track of Detailed Metadata (More Than Others)
2. Branching is Encouraged (Which Modularizes Development)
3. GitHub Has a Great Social Community

Why Git and GitHub?

Full Disclosure...

1. It Isn't the Best for Binary Files
2. GitHub Distinguishes Between Public and Private Repos

Thank you for your attention.