Version Control Introduction

Troy C. Haskin

University of Wisconsin–Madison

5/12/2014

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Paper Workflow

A typical workflow of writing a paper:

- Start with an idea/outline

- Make a draft

- Proof

- Edit

- Repeat until done

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Motivation
○○●○○○○○○
Paper Example

Version Control System
○○○○○○○○○

Git and GitHub

Workflow Shortcomings

Individual paper:

- Forget what changes were and were not made

- Make big changes but like the way it was

- Want to use the material again but alter for a different audience, journal, etc.

Group paper:

- Don't know what changes were and were not made

- Not sure what version of the paper you or others have

- Not sure who or what was added to the version in-use

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Solution

A common solution not using a version control system (VCS):

- Make a new directory and name appropriately
    - `NuclearEngineeringAndDesign`
    - `AnnalsOfNuclearEnergy`
- Make a new copy of the file and name it something different
    - `AwesomePaper-Draft1.docx`
    - `AwesomePaper-AdvisorsNotes.docx`
    - `AwesomePaper-Draft2NeedCitations.docx`
    - `AwesomePaper-Final.docx`
    - `AwesomePaper-FinalAdvisorNotes.docx`
    - `AwesomePaper-FinalFinal.docx`

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Solution Shortcomings

- Proliferation of files and directories

- No automatic list of changes; "proper" naming attempts to correct this (e.g., Draft2NeedCitations)

- Ability to go back to an earlier version would complicate naming

- Collaboration issues still not addressed

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

RELAP/MELCOR Inputs:

- Build the model by slowly adding control volumes and heat structures

- Adjust geometry input as more information becomes available

- Correct issues as they're discovered

- Might break things and need to find a older, working version

- **Re-use the input for multiple different simulations or numerical experiments**

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Writing Programs

- Start simple and add more functionality

- Fix bugs as they're discovered

- Might break things and need to find a older, working version

- Someone else might want to leap off of the work already done but apply it differently

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Same Problems

- These examples, and many more, all have the problems presented by the paper example.

- The problems only become worse as the work becomes larger or more people become involved.

What's the solution?

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Motivation
00000000
Introduction

Version Control System
●00000000

Git and GitHub

What is it?

Definition A system that records changes to a file or set of files over time so that you can recall specific versions later. [src]

Features:

- Revert a file or an entire project back to a previous state
- Review changes made over time
- See who last modified something
- Create an off-shoot from a current project state (branching)
- Create a brand new project from a current project (forking)
- Work locally and save to an online system (distributed systems)

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Advantages / Disadvantages

Advantages

- History of the project is automatically cataloged

- All versions of the project are saved and ID-ed automatically

- Line-by-line and person-by-person reviewable history.

Disadvantages:

- Can't see line-by-line changes for binary files (e.g., `*.docx` or `*.pdf`)

- Not good for saving humongous files (large binary data files shouldn't be versioned)

- Requires discipline and effort to log and sync changes

- Becomes much, much more complicated for larger projects (not a worry for us)

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

Definitions (Examples to follow)

Repository   A directory that hold all project files and VCS information.

Commit   A submission changes from the user to the VCS; this creates a new version and save the previous state earlier in the history

Commit Message   A short/long description of the changes present in the commit.

Branch   A new, separate line of history starting from a certain version; changes cane be made to a branch without affecting what it was bracnhed from

Diff   A comparison of two files with line-by-line difference highlighted

Sync/Push   A synchronization of a local repository with a non-local one

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

History/Commit Example

Create repository called
"Notes".

Notes

Notes.txt | REPO Info.

Add a new file Notes.txt
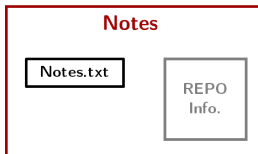
Notes
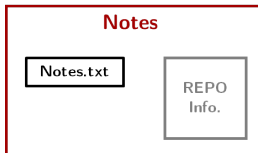
Notes.txt | REPO Info.

History:

REPO Info. initially empty

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

History/Commit Example

Commit new file to VCS.

History:
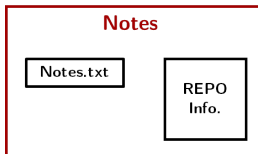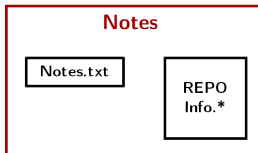
1. 'Notes.txt' created.

Add a new note to file.

History/Commit Example

Commit new line to VCS.
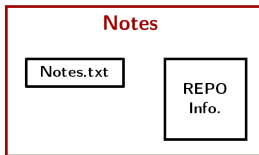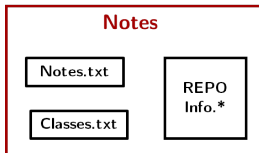
History:

1. 'Notes.txt' created.
2. Added new note to 'Notes.txt'.

Add a new file.

History/Commit Example

Commit new file to VCS.
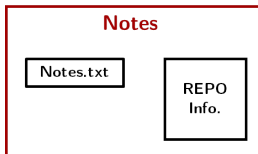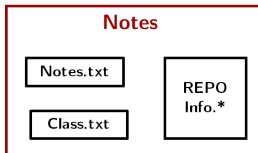
History:

1. 'Notes.txt' created.
2. Added new note to 'Notes.txt'.
3. Added new file 'Classes.txt'.

And so on ...

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

History of This Presentation

History

Branching Example

Branches share a common ancestor but can have different histories after
the branch commit.
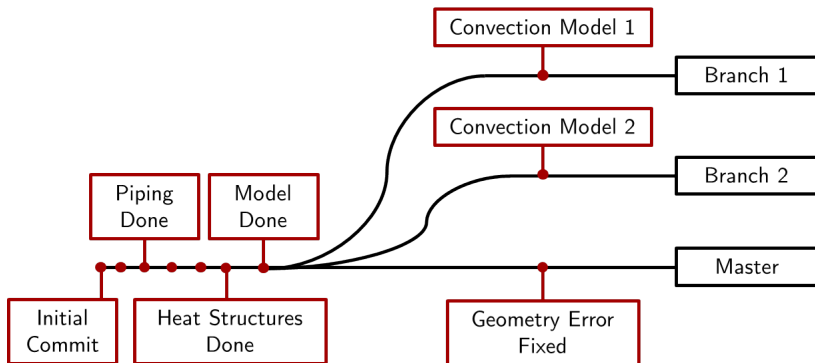
Merge Example

It is possible to merge histories in branches but can lead to conflicts (mismatched or ambiguous histories that require resolution).

That's it

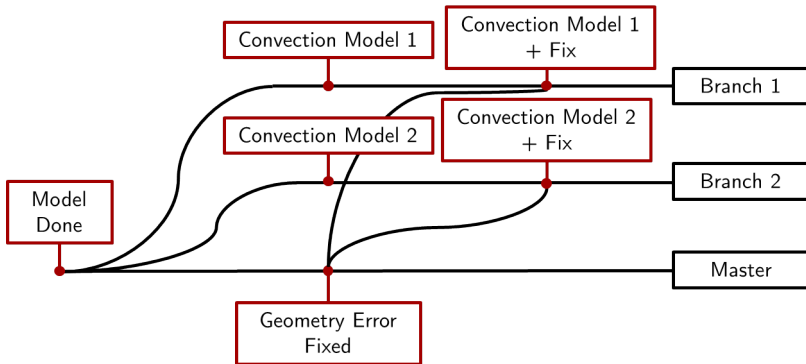And that covers the broad introduction.

There is more, of course, but that will be investigated later.

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

DEPARTMENT OF

ENGINEERING PHYSICS

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

The Program and the Website

Git My VCS program of choice; created to manage one of the largest collaborative projects in history — the Linux Kernel.

GitHub A website the allows online hosting of repositories and uses Git as its VCS. Public repositories are completely free; private repositories cost money.

DEPARTMENT OF
ENGINEERING PHYSICS
COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON