

# Version Control Introduction

Troy C. Haskin

University of Wisconsin–Madison

5/12/2014

DEPARTMENT OF

**ENGINEERING PHYSICS**

COLLEGE OF ENGINEERING UNIVERSITY OF WISCONSIN-MADISON

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 Git and GitHub

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 Git and GitHub

# Paper Workflow

A typical workflow of writing a paper:

- Start with an idea/outline
- Make a draft
- Proof
- Edit
- Repeat until done

# 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

# 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

# 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

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 Git and GitHub



# 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**

# 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

# 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?

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 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)

# 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)

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 Git and GitHub

## 1 Motivation

Paper Example

Other Examples

## 2 Version Control System

Introduction

How it Works

## 3 Git and GitHub