

A good idea is worthless  
without impeccable  
execution and a  
commitment to iterate.

*Startup Quote!*



**ZACH KLEIN**

*CO-FOUNDER, VIMEO*

Bronco Scheduler **Home** Registration Guide Tips & Tricks Feedback

Fall 2015 Summer 2015

Like Share Bryan Thornbury and 520 others like this.

Go to Bronco Scheduler Homepage

SCHEDULER

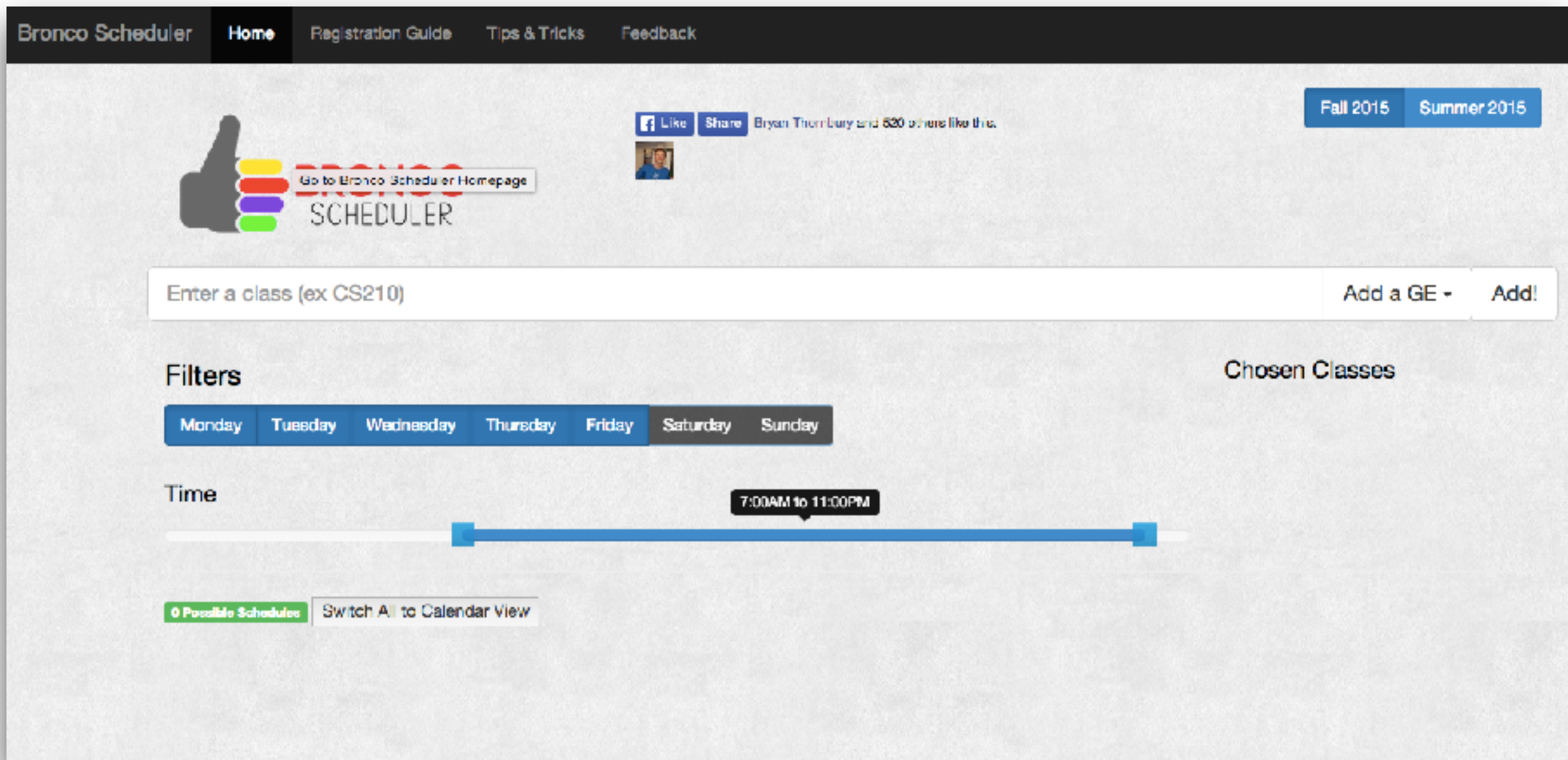
Enter a class (ex CS210) Add a GE - Add!

Filters Chosen Classes

Monday Tuesday Wednesday Thursday Friday Saturday Sunday

Time 7:00AM to 11:00PM

0 Possible Schedules Switch All to Calendar View



# Bronco Scheduler

<http://brancoscheduler.com/>



# Version Control

## - A Brief Introduction to Git

CS480 Software Engineering

Yu Sun, Ph.D.

<http://yusun.io>

[yusun@cpp.edu](mailto:yusun@cpp.edu)



CAL POLY POMONA

# Why Version Control?



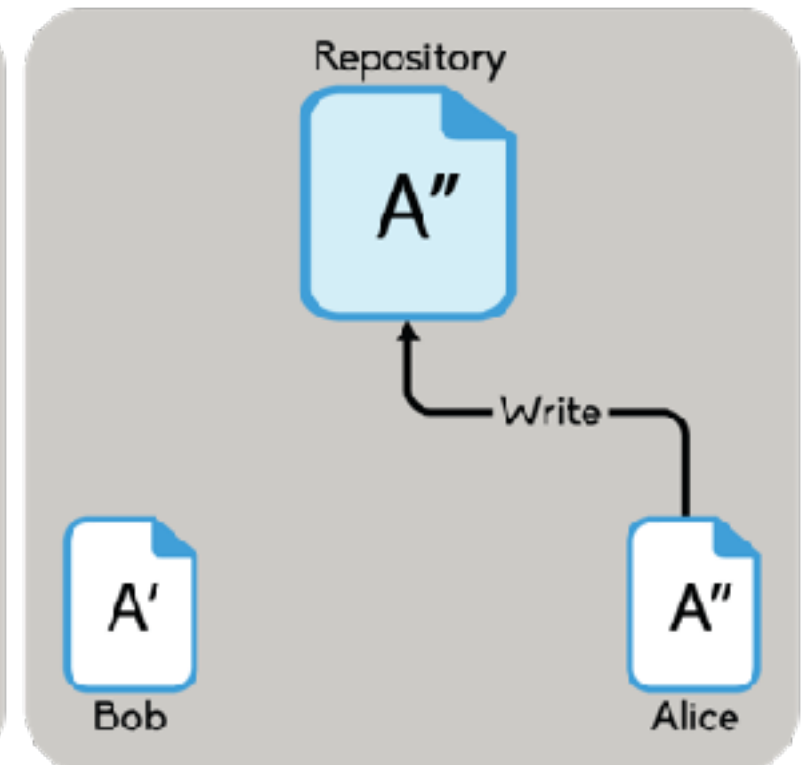
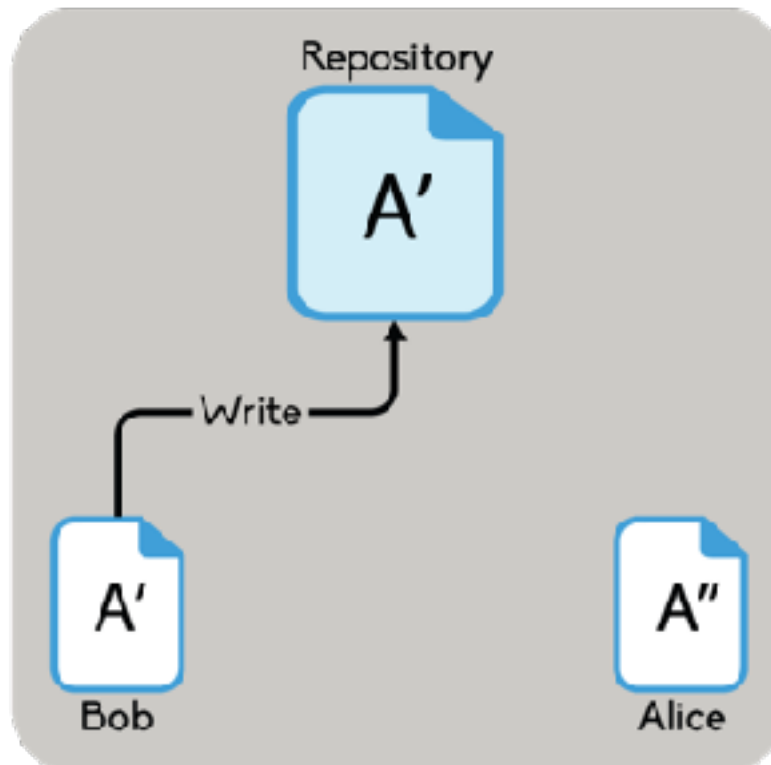
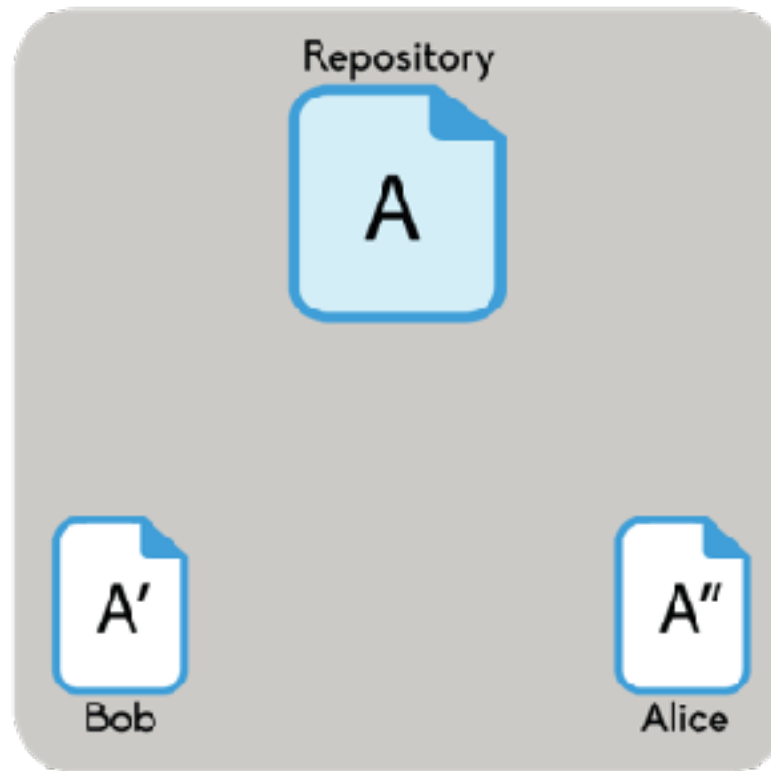
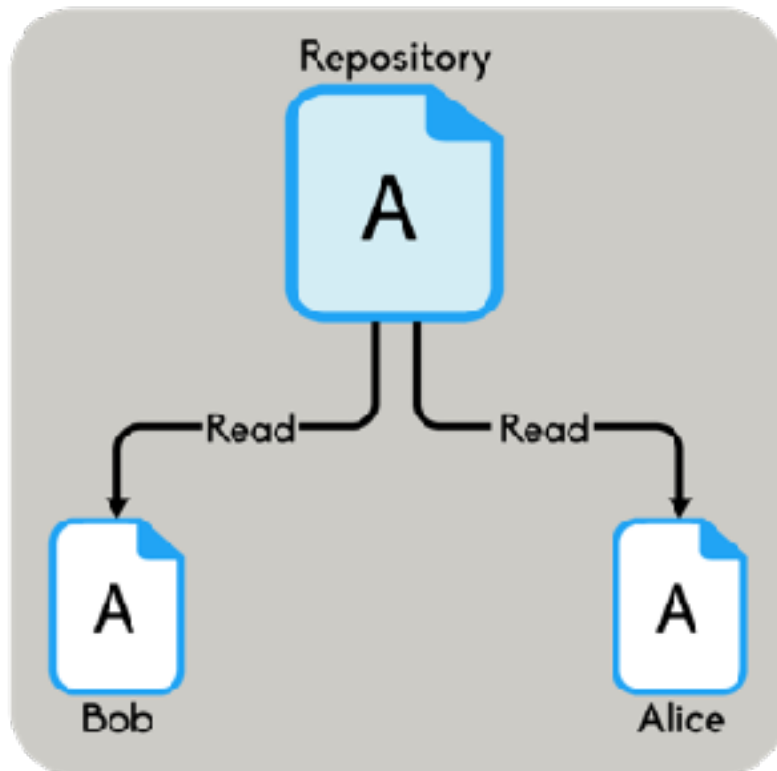
# Maintain Multiple Versions

- Safe backup
- Change version



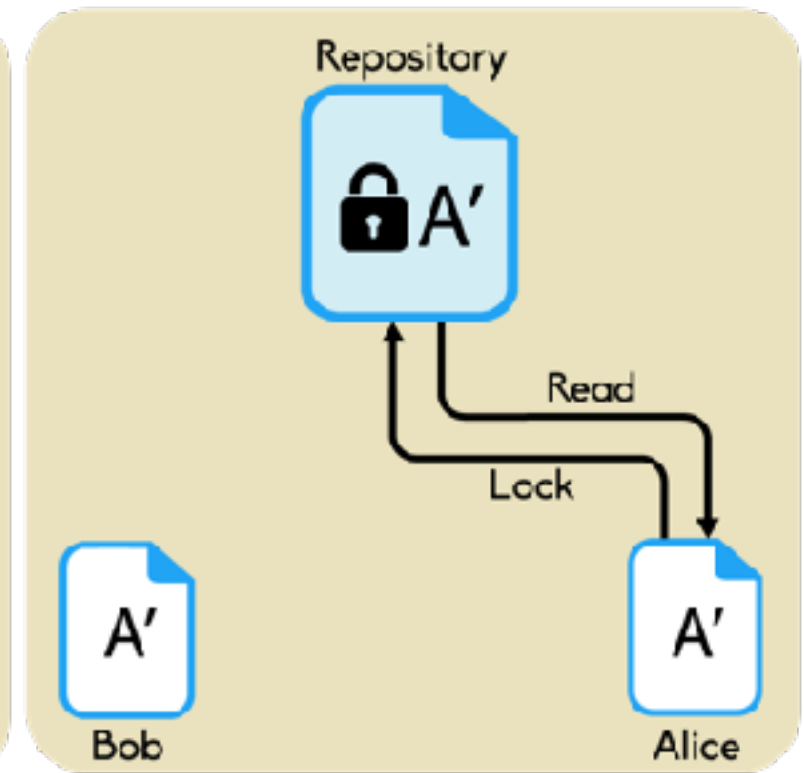
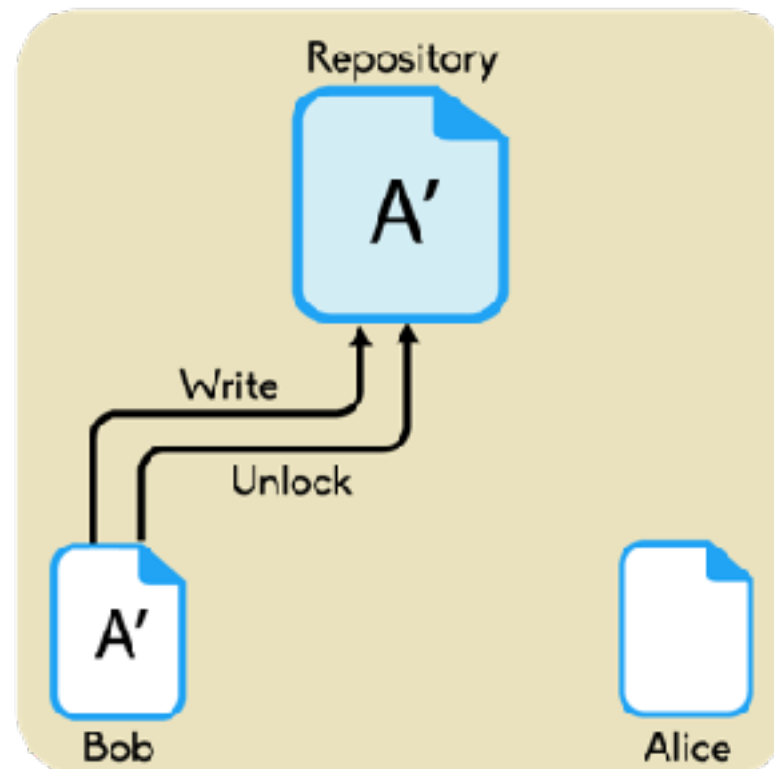
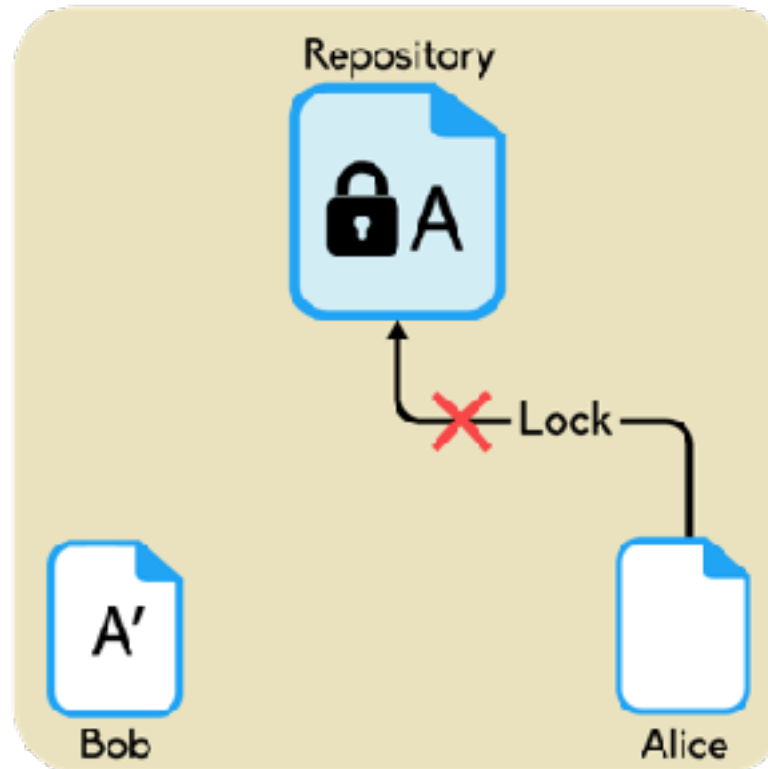
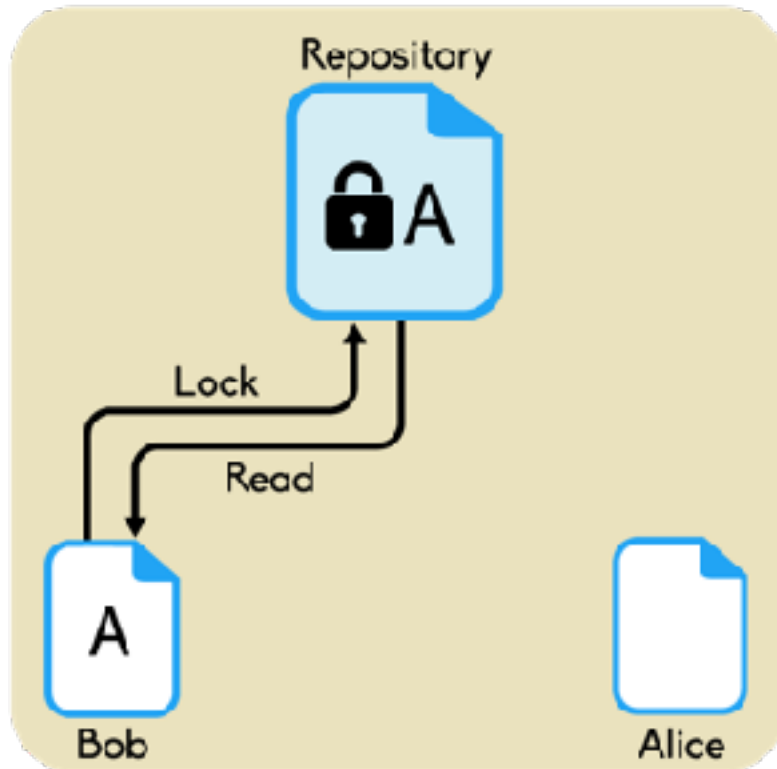
# Collaboration

The problem to avoid



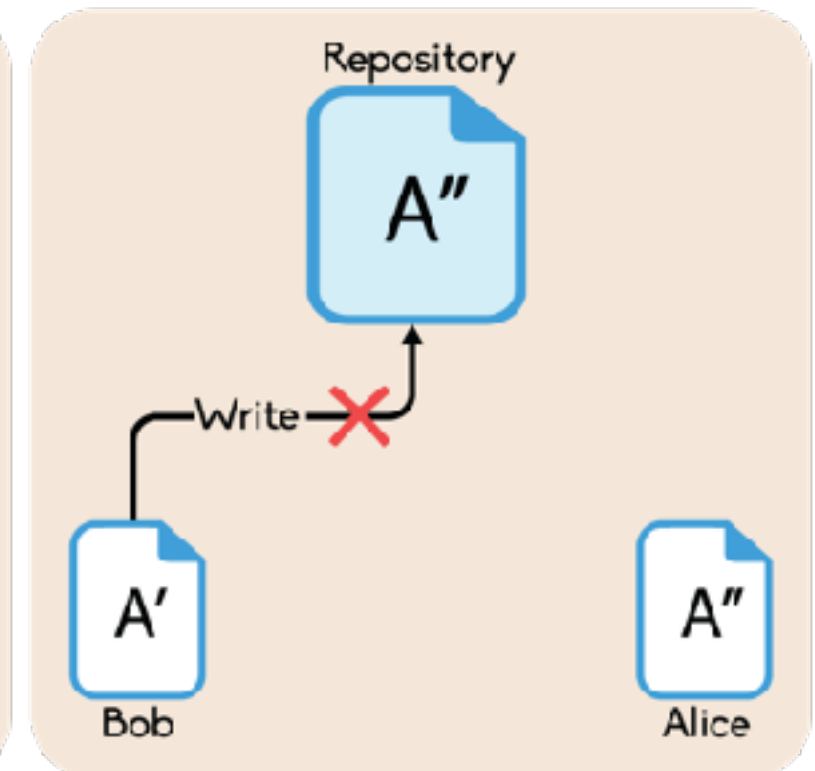
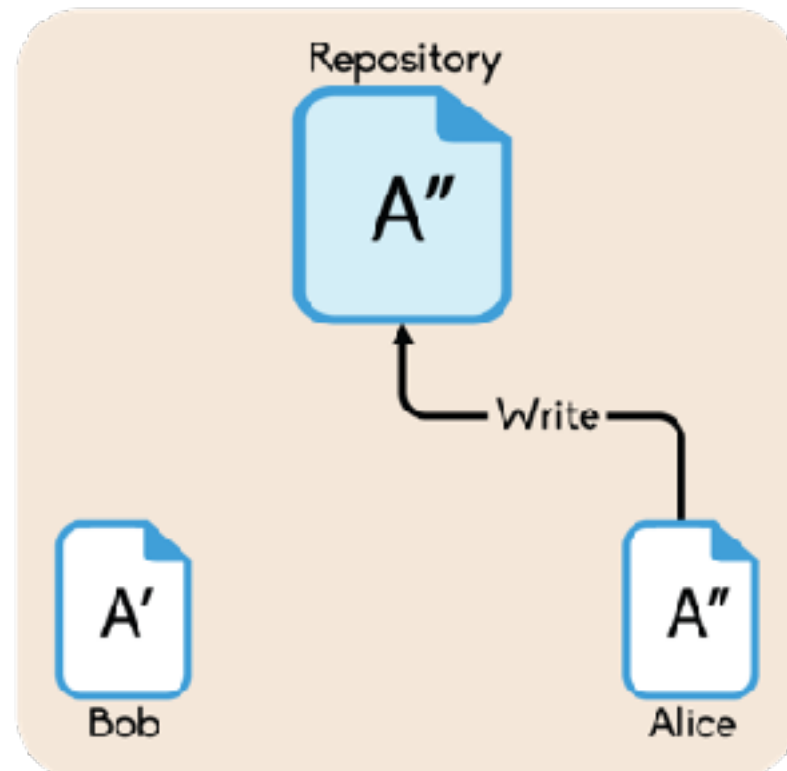
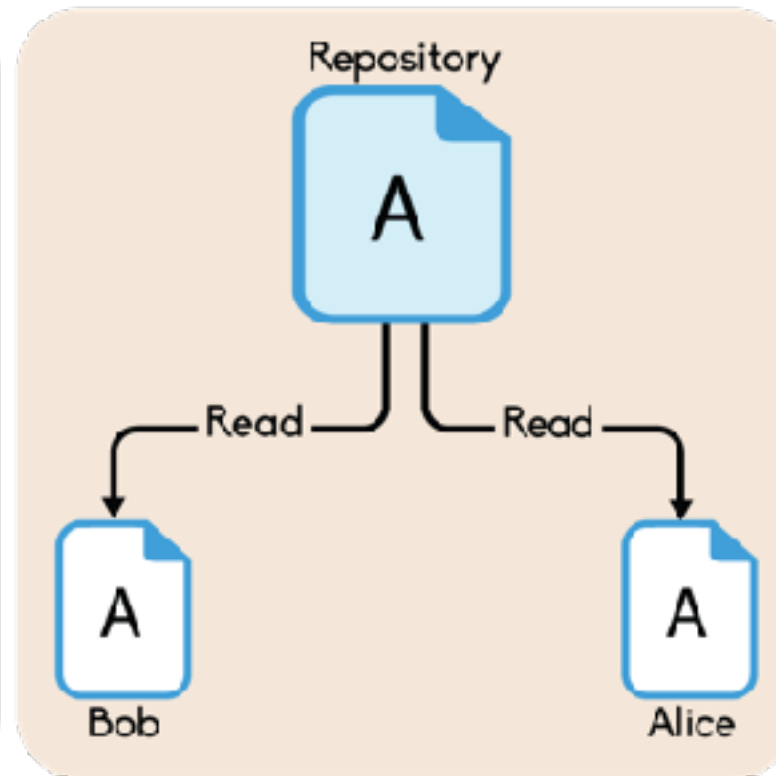
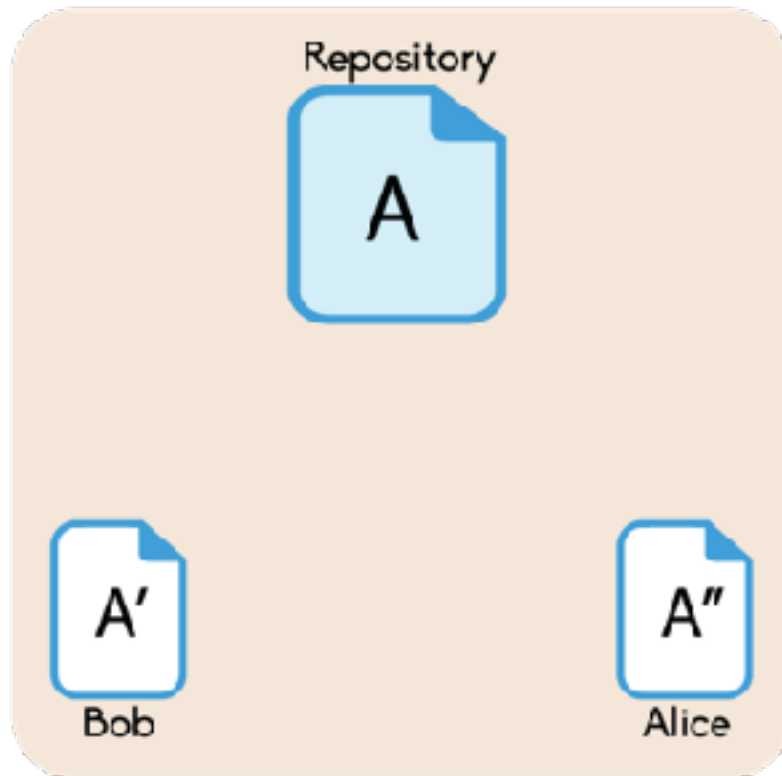
# Collaboration

## The lock-modify-unlock solution



# Collaboration

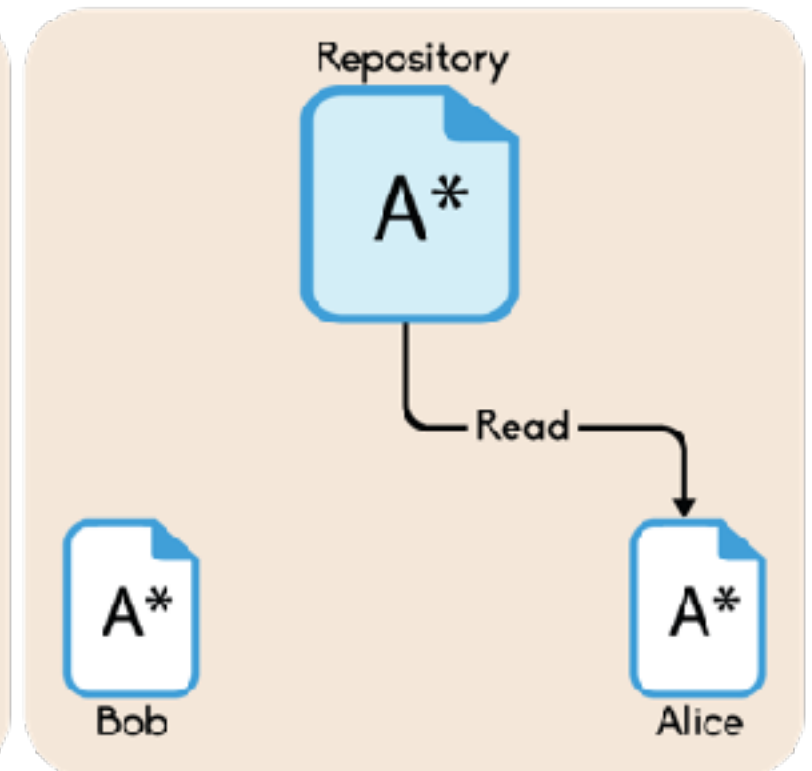
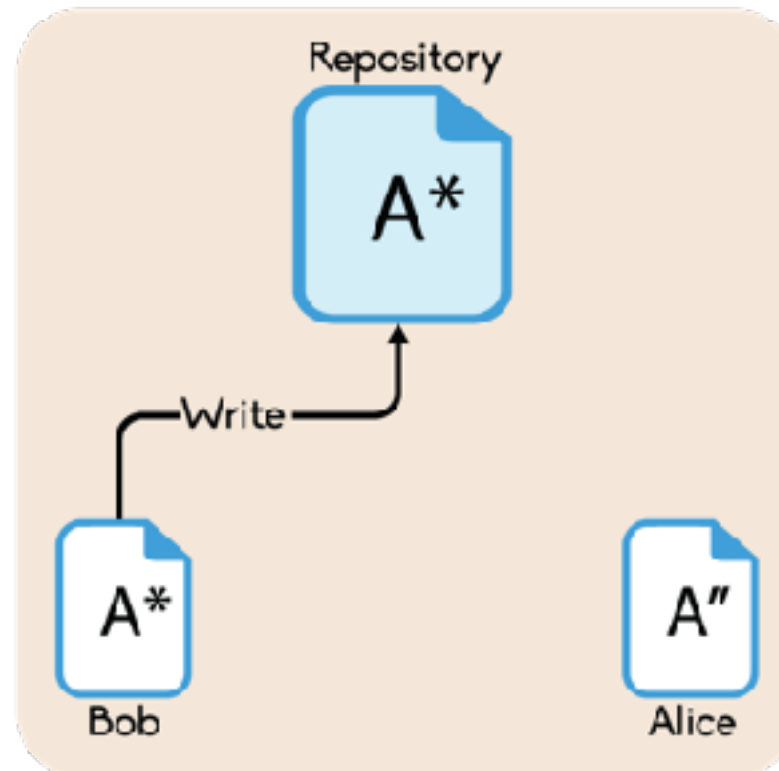
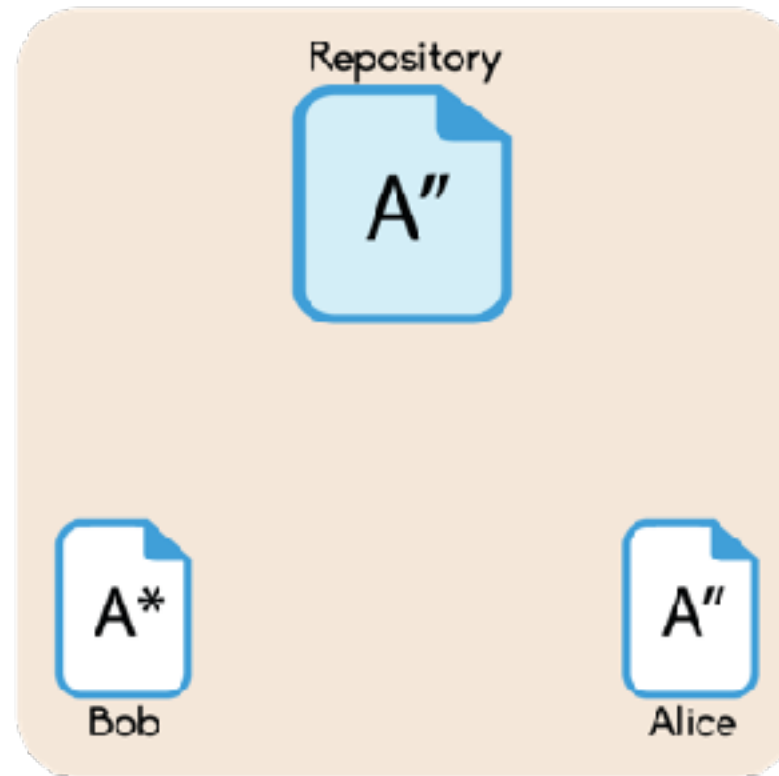
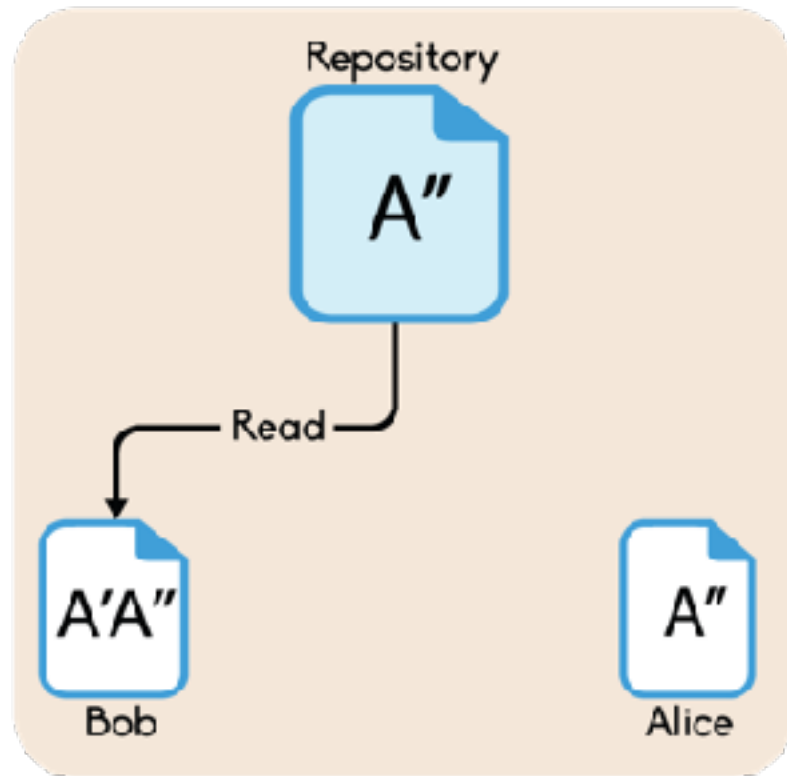
## The copy-modify-merge solution





# Collaboration

## The copy-modify-merge solution



# Monitor and Track Progress

■ = added, ■ = deleted, ■ = changed,

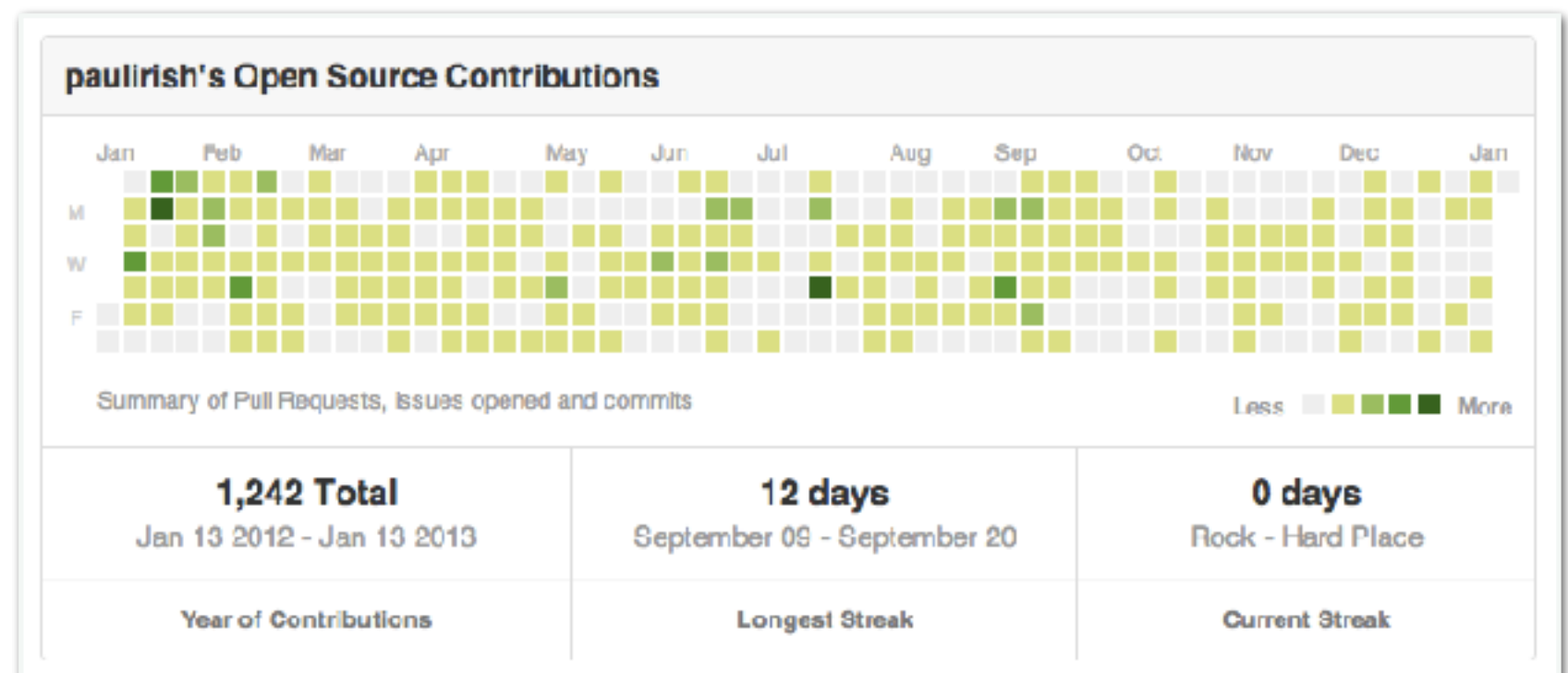
```
 HelloWorld.cs (revision 24)
01 // HelloWorld
02 public class Hello1
03 {
04     // I am adding this line so that I
05     public static void Main()
06     {
07         System.Console.WriteLine("Hello,
08     }
09 }
10

 HelloWorld.cs (revision 25)
01 // HelloWorld
02 public class Hello1
03 {
04     public static void Main()
05     {
06         System.Console.WriteLine("Hello,
07     }
08     // Adding here
09 }
10
```

Code Difference



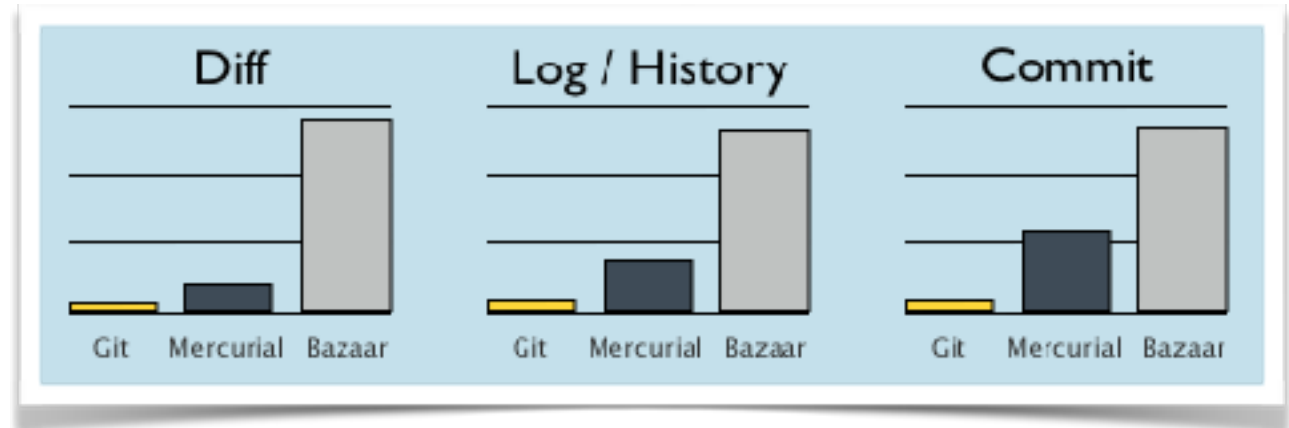
Code Contribution



**Why Git?**

# Why Git?

- Performance
- Github
- Popular



## Companies & Projects Using Git

Google

facebook

Microsoft

twitter

LinkedIn

NETFLIX



PostgreSQL



# Git History







"I'm an egotistical  
bastard, and I name  
all my projects after  
myself.

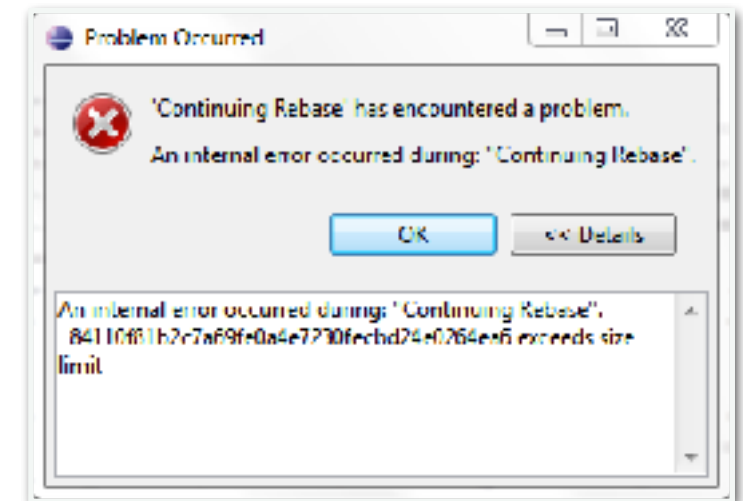
First *Linux*, now *git*."

— Linus Torvalds

# Why Command-Line?

# Why Git Command-Line?

- Graphical clients are based on CLT
- Graphical clients could cause problems
- Integrated with shell scripts
- Graphical clients not always available



```
#!/bin/bash
# TANER'S DEVELOPMENT SERVER BACKUP SCRIPT
# BACKUP CVS, BUG TRACKING and WEBSITE with one command
# SCRIPT MUST BE RUN BY USER TANER
# Prerequisites: directory /home/taner should exist
# Author: Taner

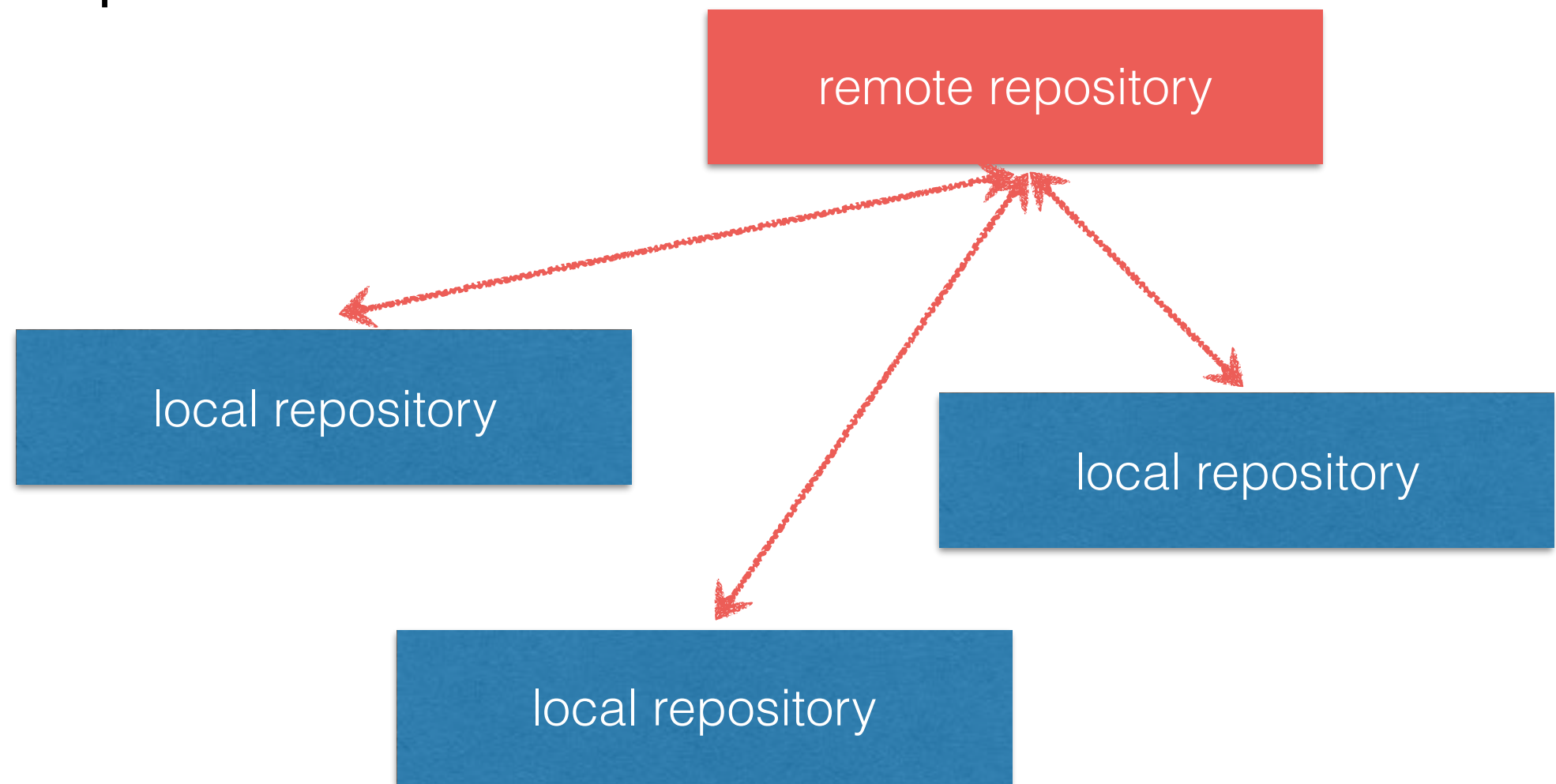
echo
echo "This script will backup the internal website, "
echo "cvs repository and bug database"
choice=""

# This function is simply to get a yes or no from the user
# keeps looping until the user enters a valid value
inputYesOrNo() {
    choice=""
    read choice
    if [ -z "$choice" ]
    then
        inputYesOrNo
    fi
    if [ $choice = 'y' ] || [ $choice = 'Y' ]
    then
        choice='y'
    fi
    if [ $choice = 'n' ] || [ $choice = 'N' ]
    then
        choice='n'
    fi
    if [ $choice != 'n' ] && [ $choice != 'y' ]
    then
        echo "Please enter 'y' or 'n'"
        inputYesOrNo
    fi
}
```

# Git Exercises

# 1. Create Git Repositories

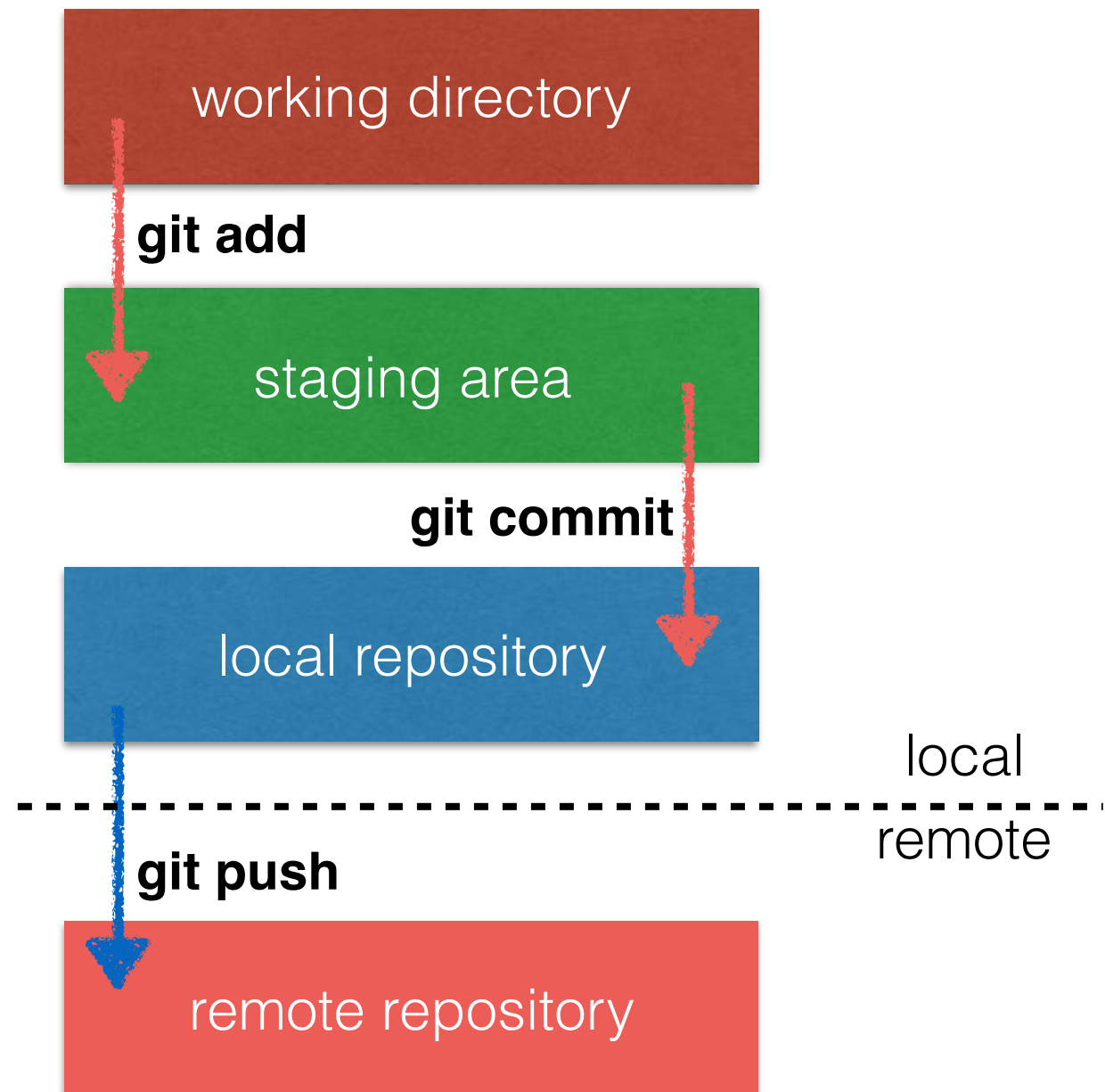
- `git clone <repo>`
- `git init <repo>`





## 2. Add/Commit/Push

- `git add <path>`
- `git commit`
- `git push`



# 3. Check Status

- `git status`
- `git log`
- `git branch`

```
|gh-pages x| → git status
On branch gh-pages
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   images/boxes.png
    new file:   images/empty.png
    new file:   images/ignored.png
    new file:   images/pallet.png
    new file:   images/push.png
    new file:   images/truck.png
    new file:   images/untracked.png

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   index.html
    modified:   init.md

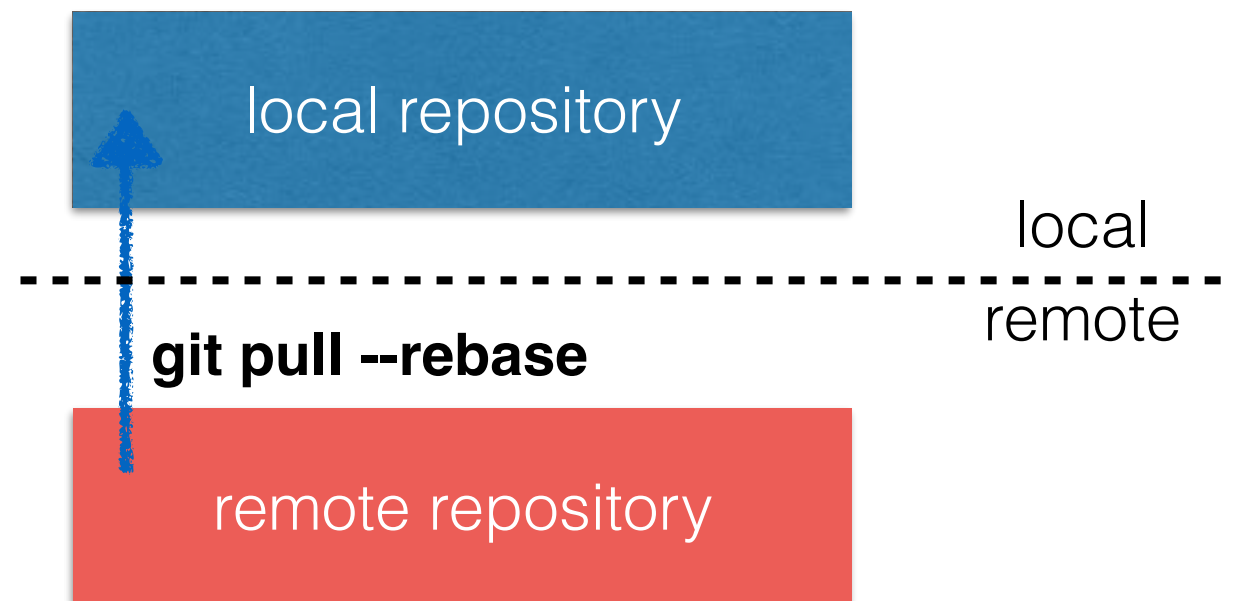
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    fork.md
```



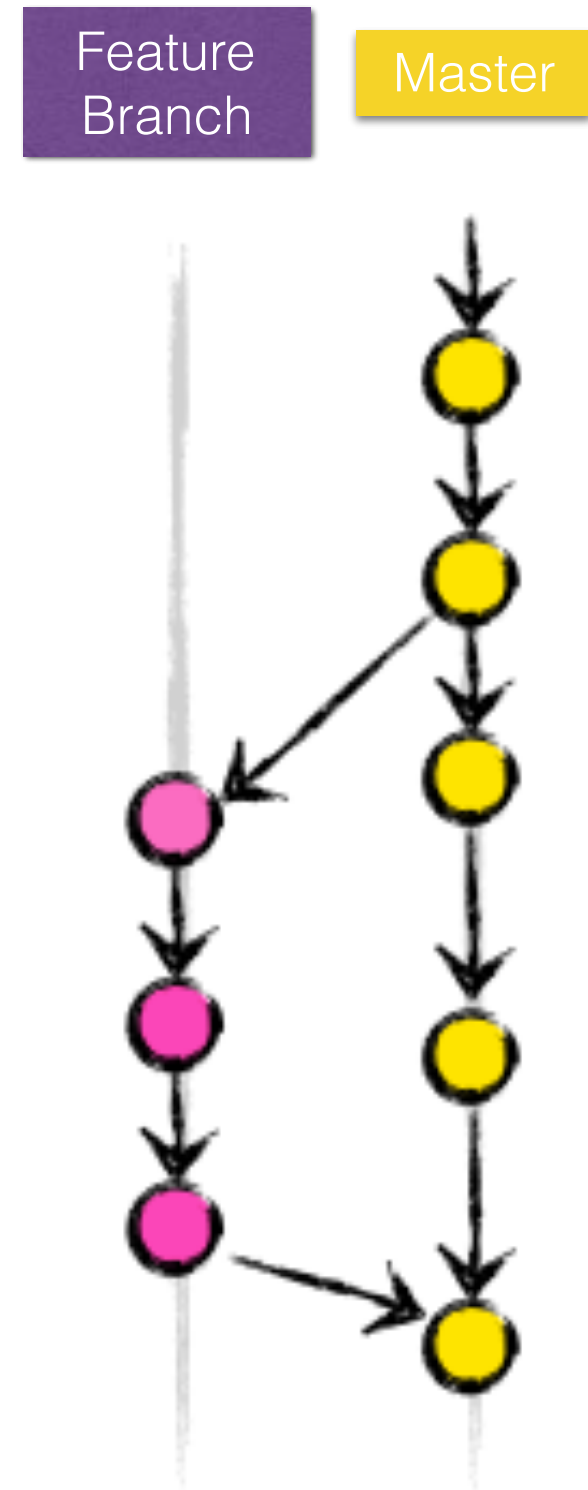
# 4. Sync Changes & Resolve Conflicts

- `git pull --rebase` (referred approach)
- `git pull`
  
- `git status`
- `// clean up conflicts`
- `git add <conflicted file>`
- `git rebase --continue`
  
- `git push`



# 5. Branches

- `git checkout -b <branch>`
- `git checkout <branch>`
- `git merge <branch>`



## 6. Git Undo

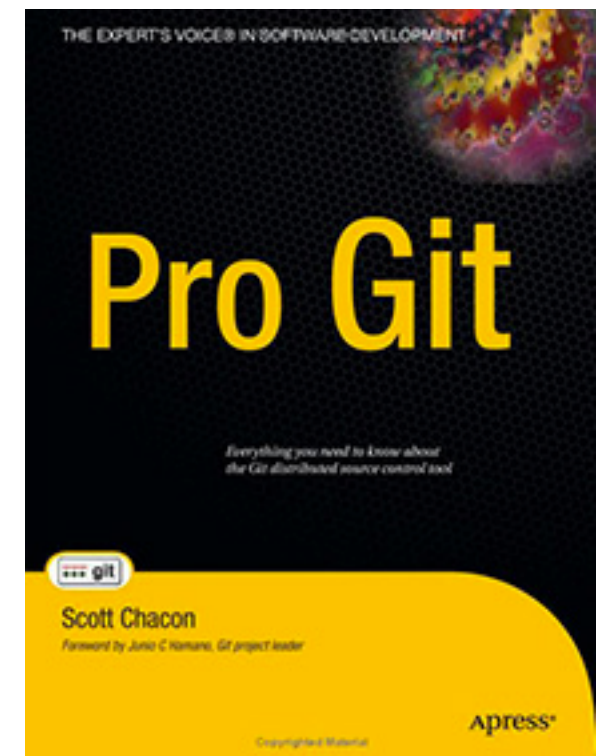
- Always backup first
- Google the solution





# Git Learning Resources

- <http://git-scm.com/>
- <https://www.youtube.com/user/GitHubGuides>
- **Google it!**



# Git Basics Overview

