Version Control
- A Brief Introduction to Git

CS480 Software Engineering
http://cs480.yusun.io
January 14, 2015

Yu Sun, Ph.D.
http://yusun.io
yusun@cpp.edu

CAL POLY POMONA

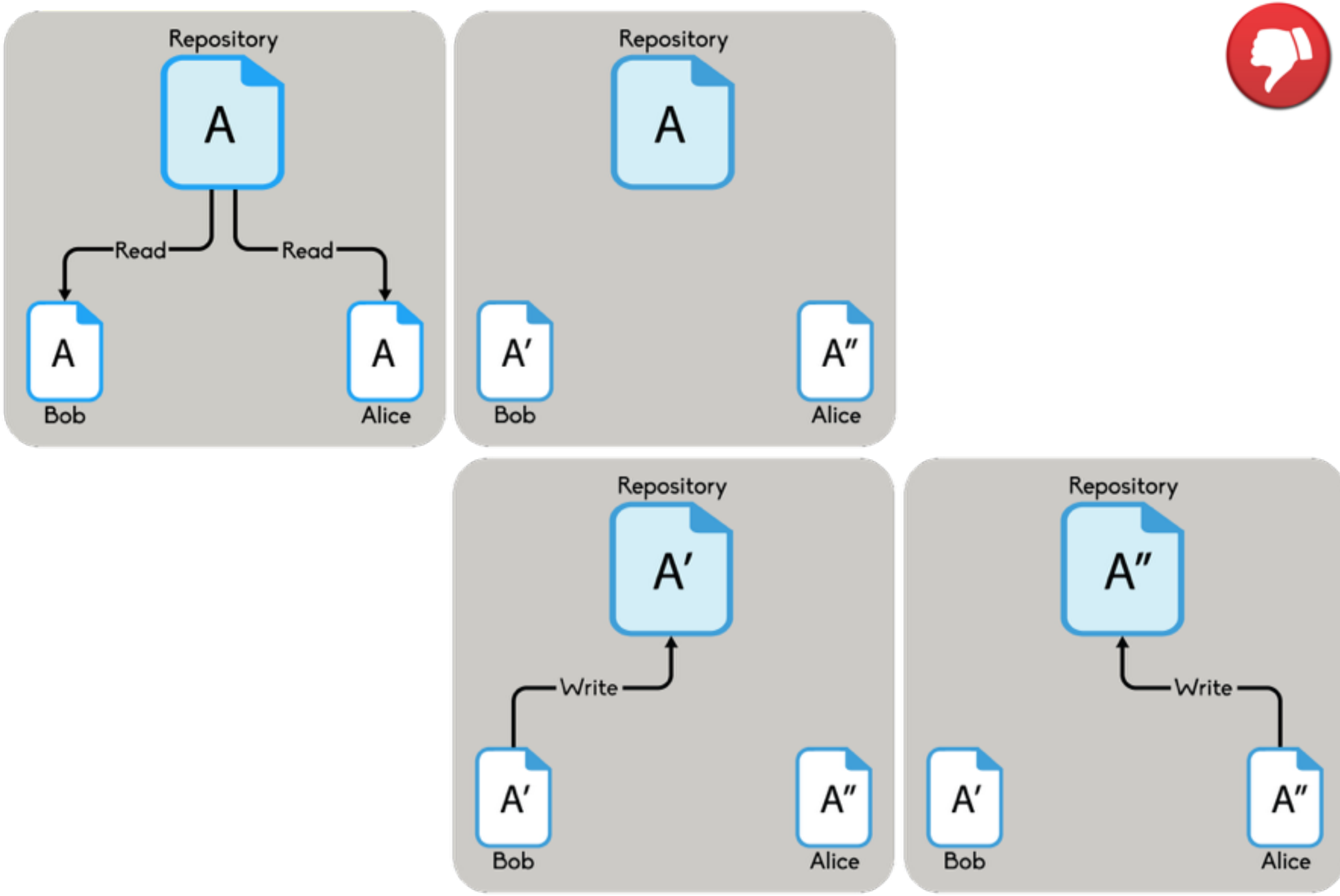
Why Version Control?

Maintain Multiple Versions

- Safe backup
- Change version

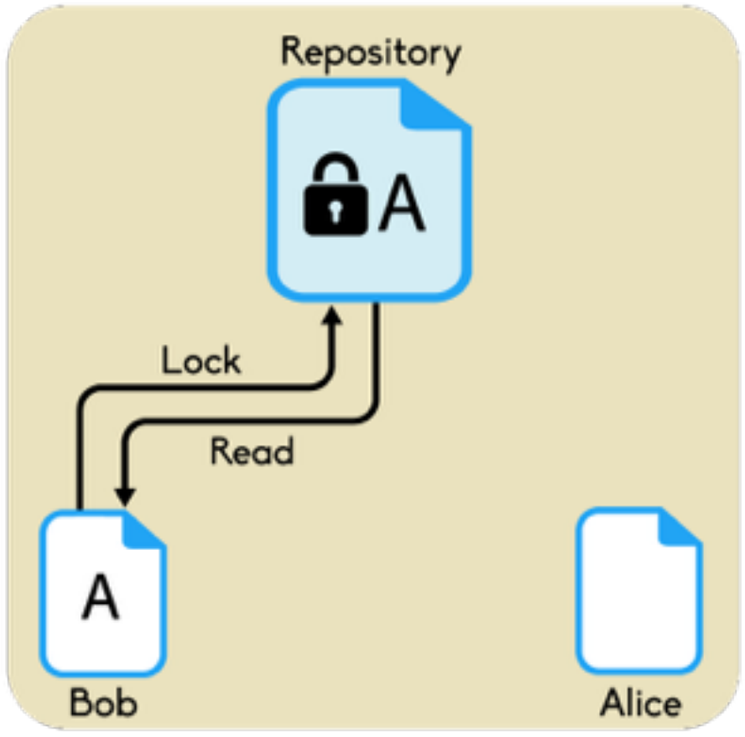
Collaboration

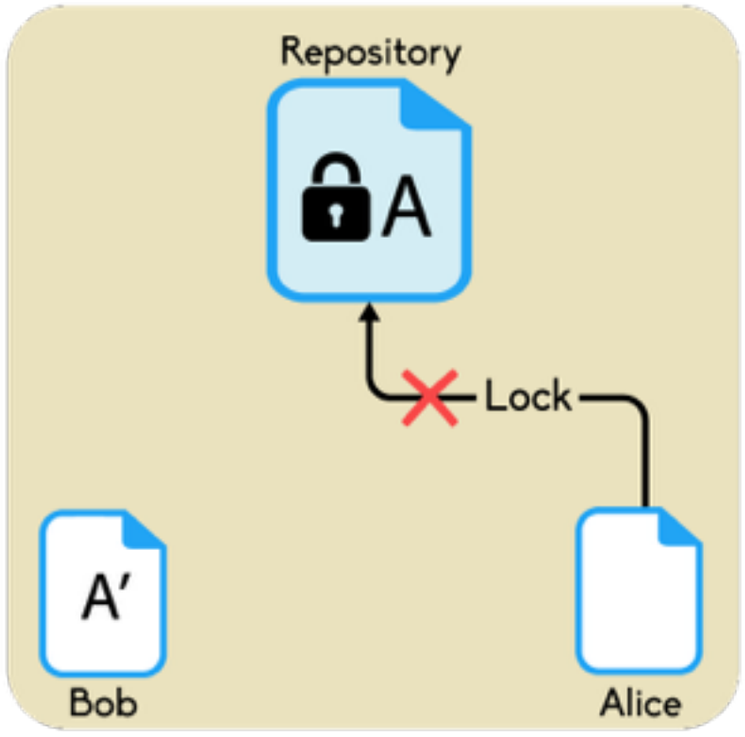
The problem to avoid

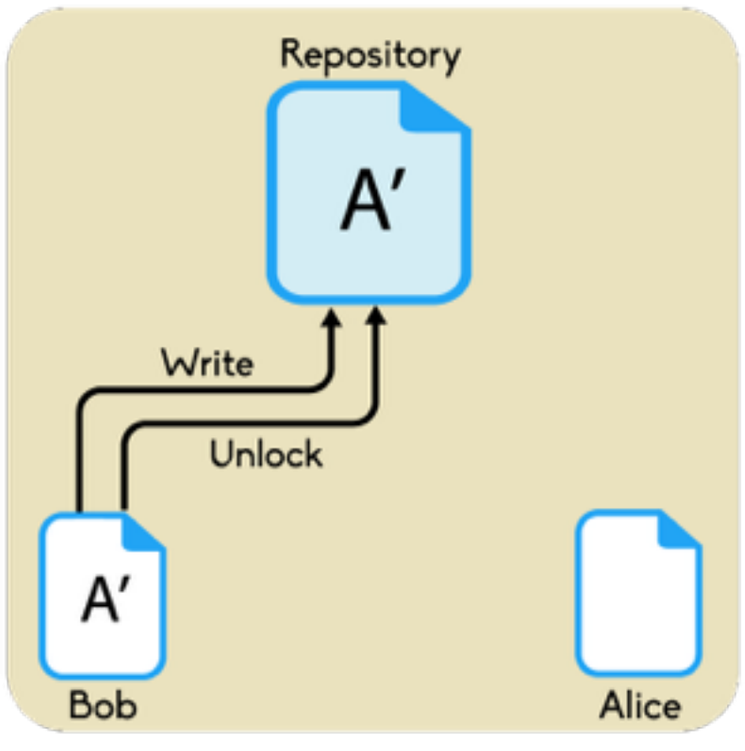


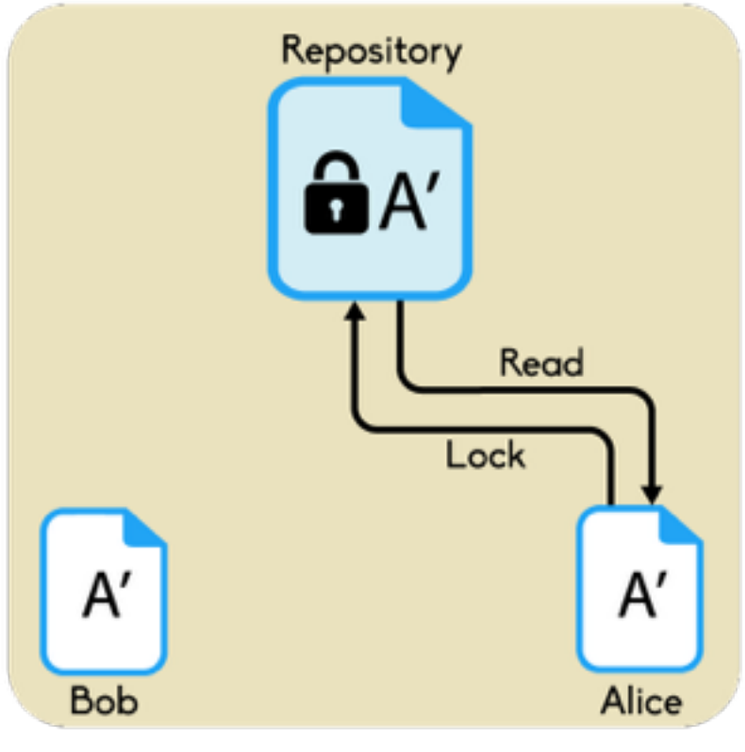
Collaboration

The lock-modify-unlock solution



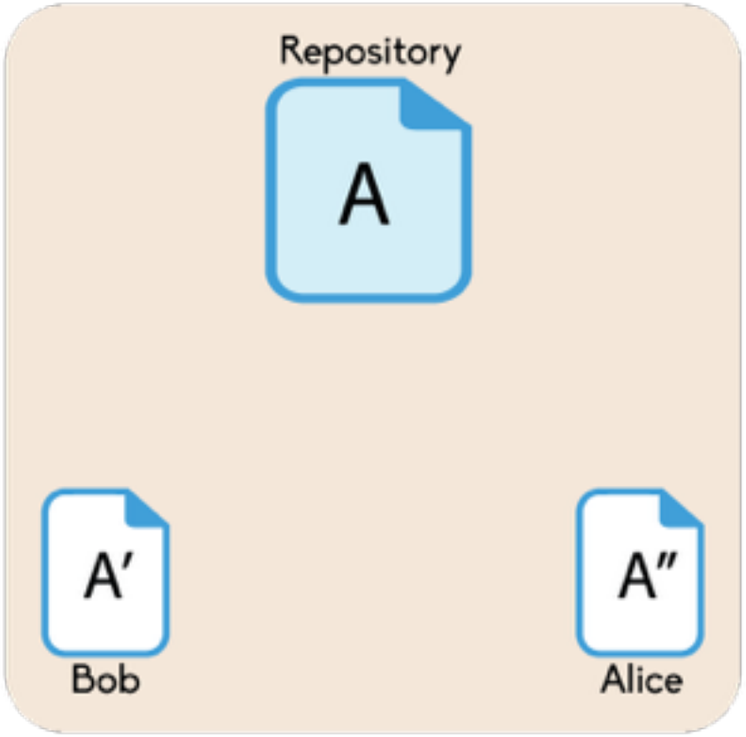


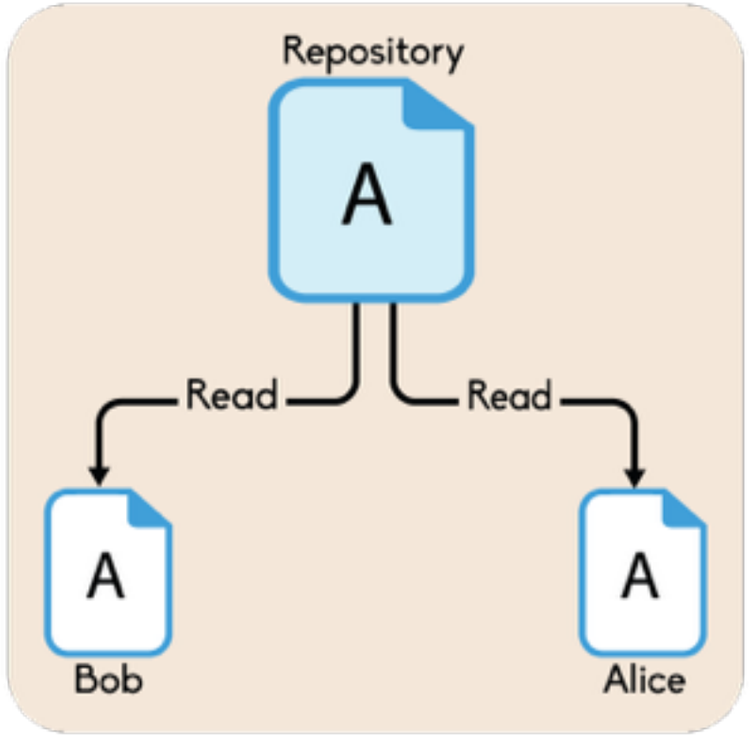


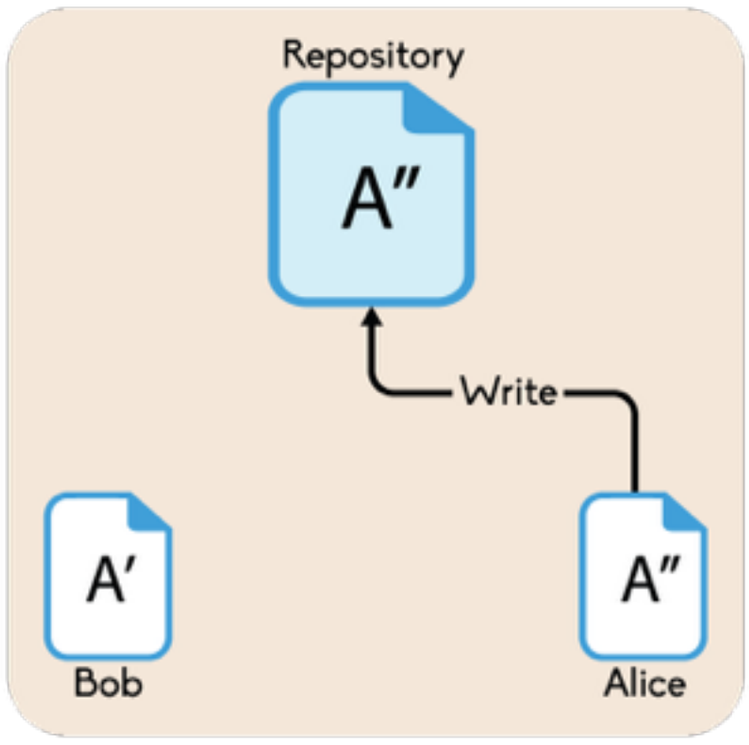


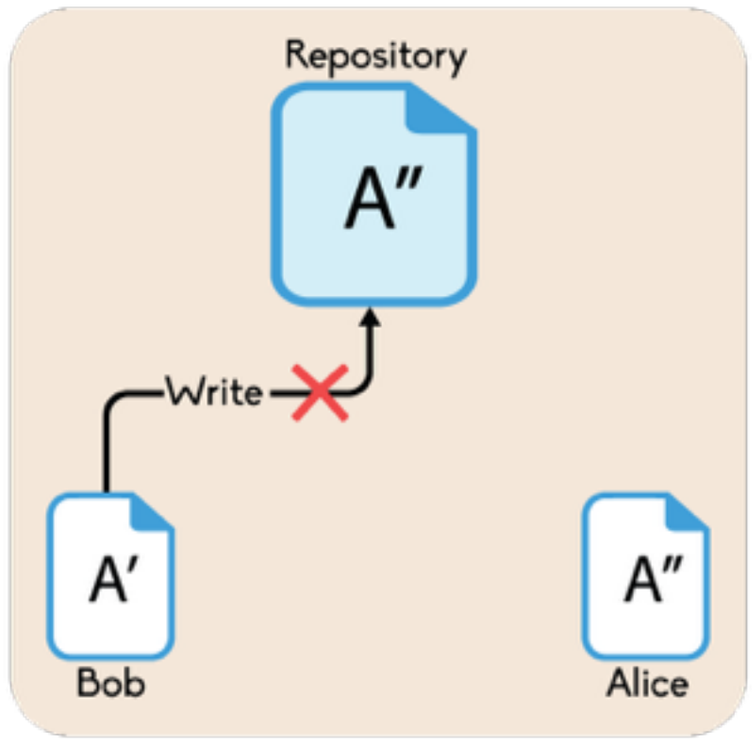
Collaboration

The copy-modify-merge solution



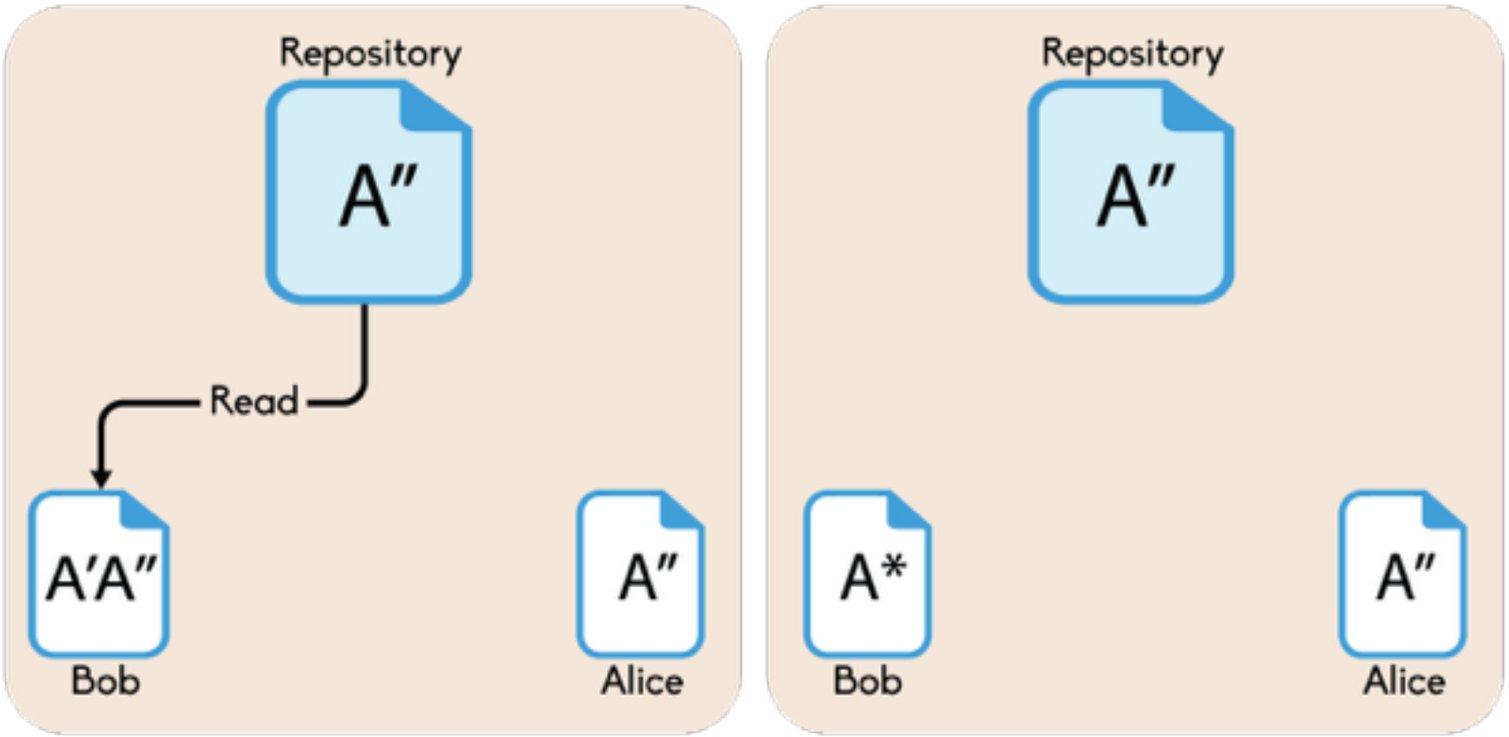


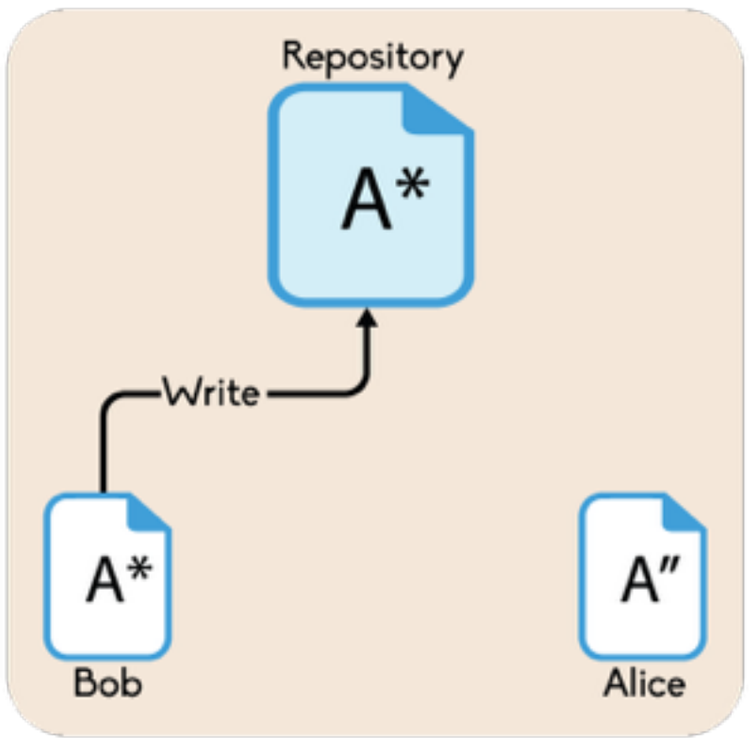


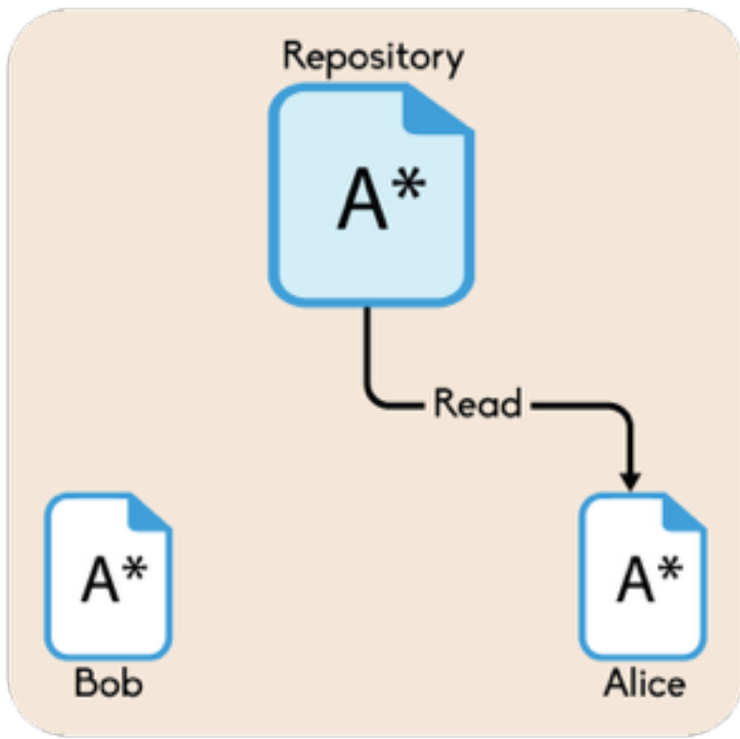


Collaboration

The copy-modify-merge solution







Monitor and Track Progress

= added, = deleted, = changed,

HelloWorld.cs (revision 24)

```
01 // Hello1.cs
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
```

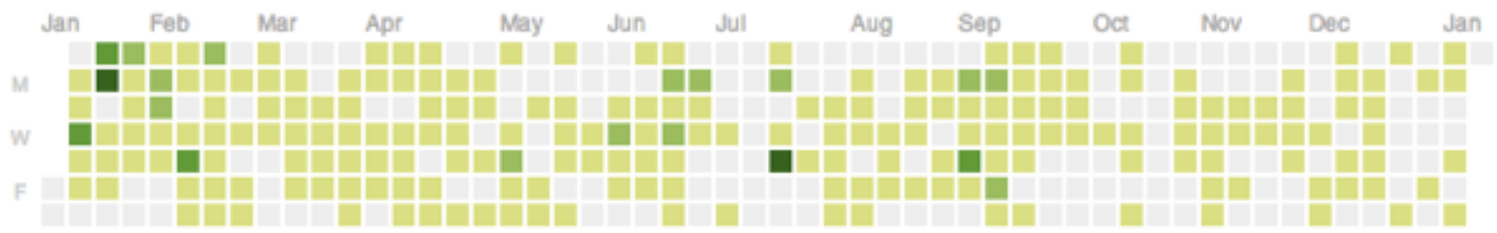
HelloWorld.cs (revision 25)

```
01 // Hello1.cs
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

paulirish's Open Source Contributions



Summary of Pull Requests, issues opened and commits

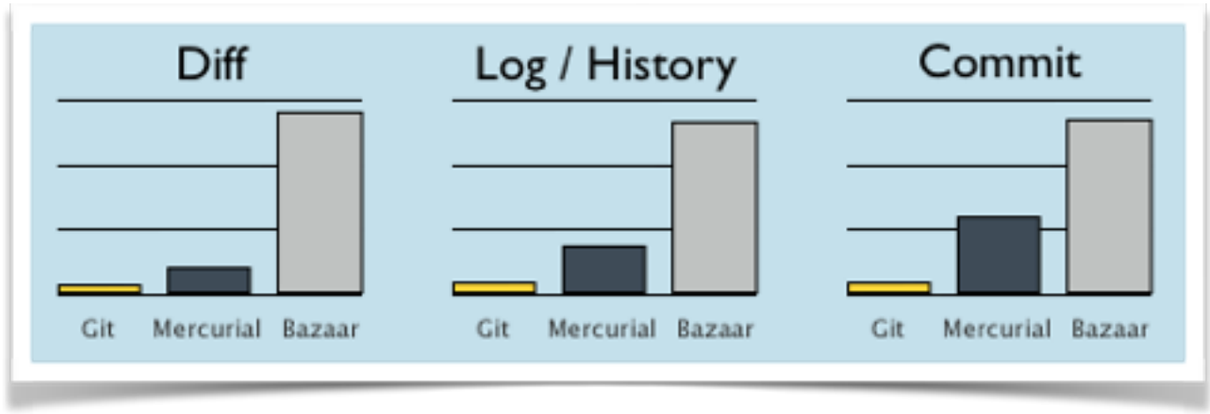
Less More

1,242 Total	**12 days**	**0 days**
Jan 13 2012 - Jan 13 2013	September 09 - September 20	Rock - Hard Place
Year of Contributions	**Longest Streak**	**Current Streak**

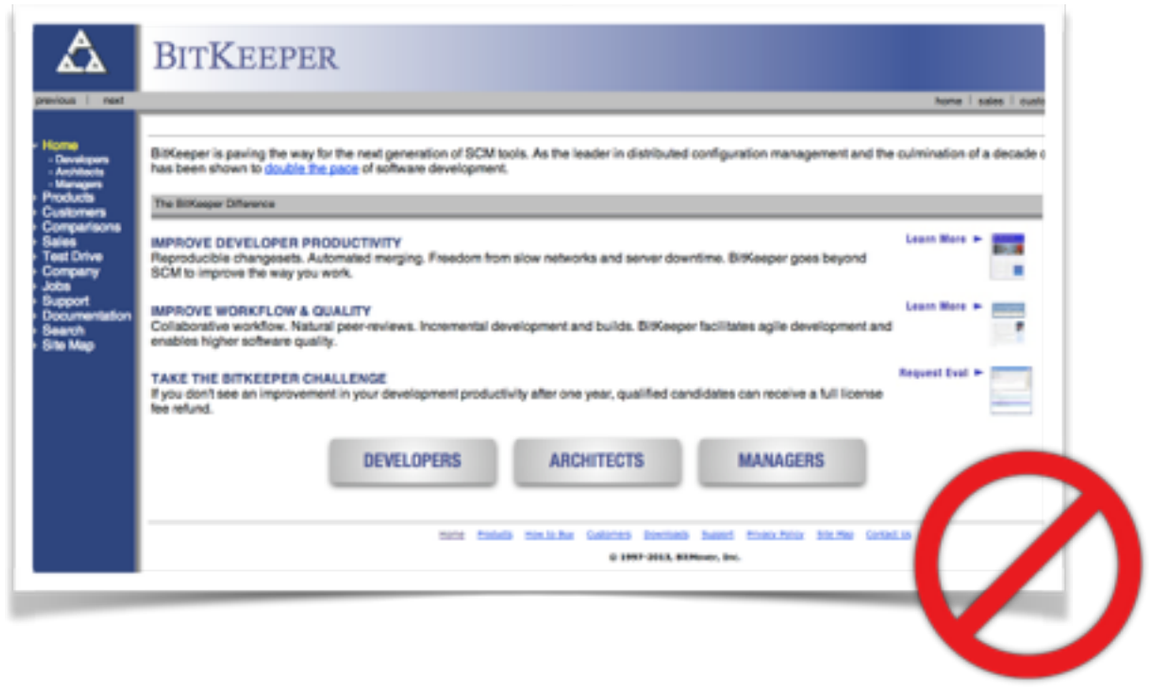
Why Git?

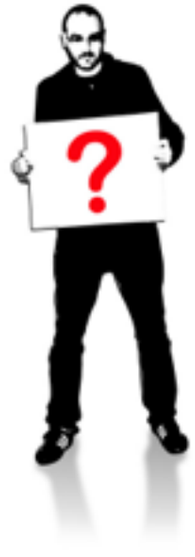
Why Git?

- Performance
- Github
- Popular



Git History


BitKeeper
Home
Developers
Architects
Managers
Products
Customers
Comparisons
Sales
Test Drive
Company
Jobs
Support
Documentation
Search
Site Map
BitKeeper is paving the way for the next generation of SCM tools. As the leader in distributed configuration management and the culmination of a decade
has been shown to double the pace of software development.
The BitKeeper Difference
IMPROVE DEVELOPER PRODUCTIVITY
Reproducible changesets. Automated merging. Freedom from slow networks and server downtime. BitKeeper goes beyond
SCM to improve the way you work.
Learn More
IMPROVE WORKFLOW & QUALITY
Collaborative workflow. Natural peer-reviews. Incremental development and builds. BitKeeper facilitates agile development and
enables higher software quality.
Learn More
TAKE THE BITKEEPER CHALLENGE
If you don't see an improvement in your development productivity after one year, qualified candidates can receive a full license
fee refund.
Request Eval
DEVELOPERS
ARCHITECTS
MANAGERS



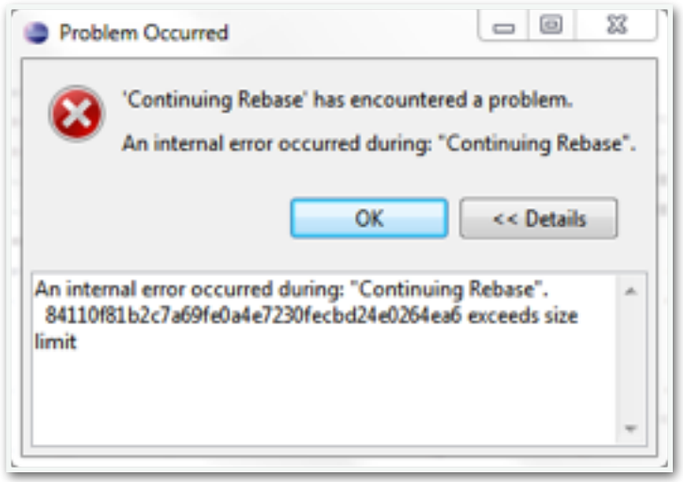
"I'm an egotistical bastard, and I name all my projects after myself.
First Linux, now git."

— Linus Torvalds

Why Command-Line?

Why Git Command-Line?

- Graphical clients are based on CLT
- Graphical clients could cause problems

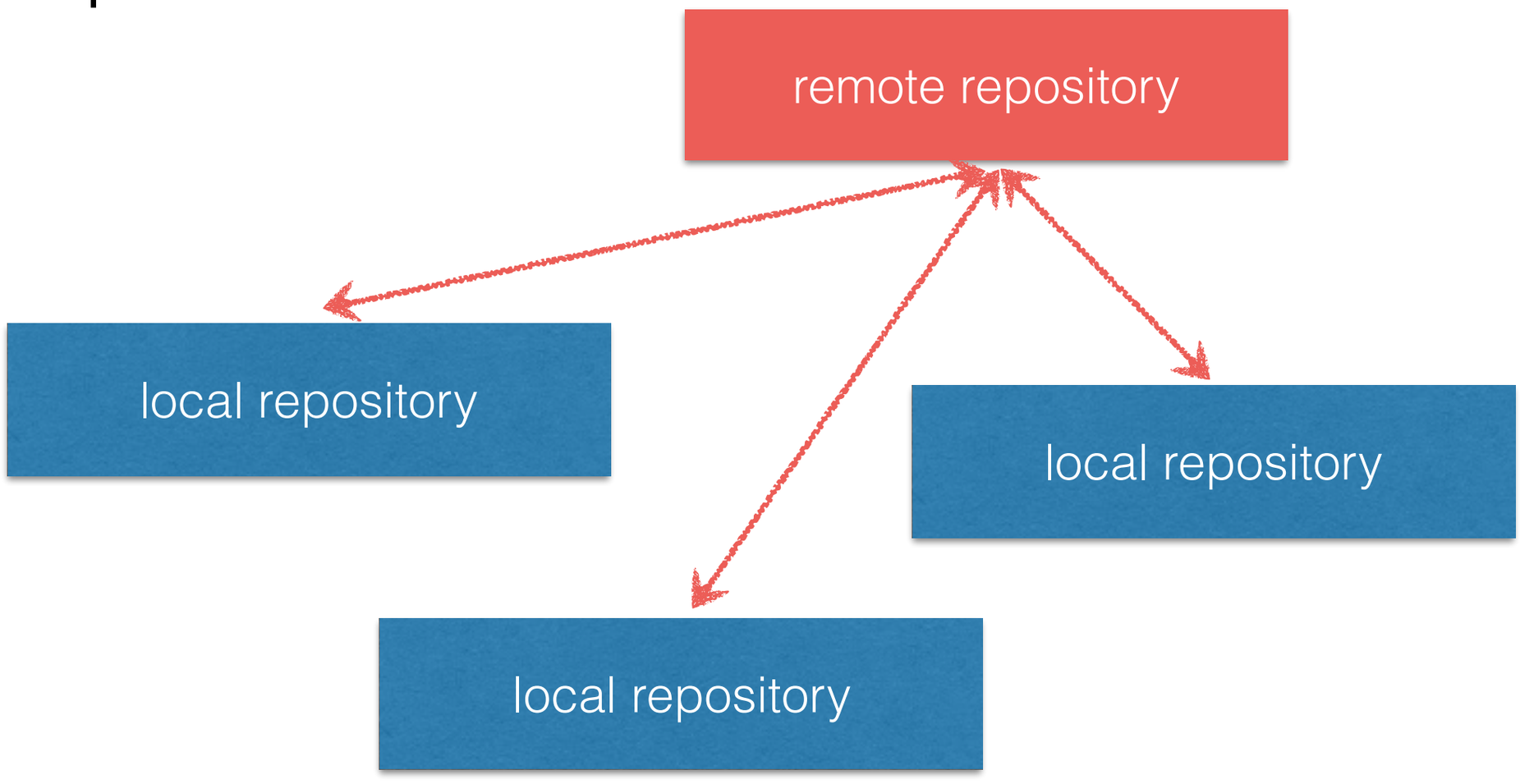


- Integrated with shell scripts
- Graphical clients not always available

Git Exercises

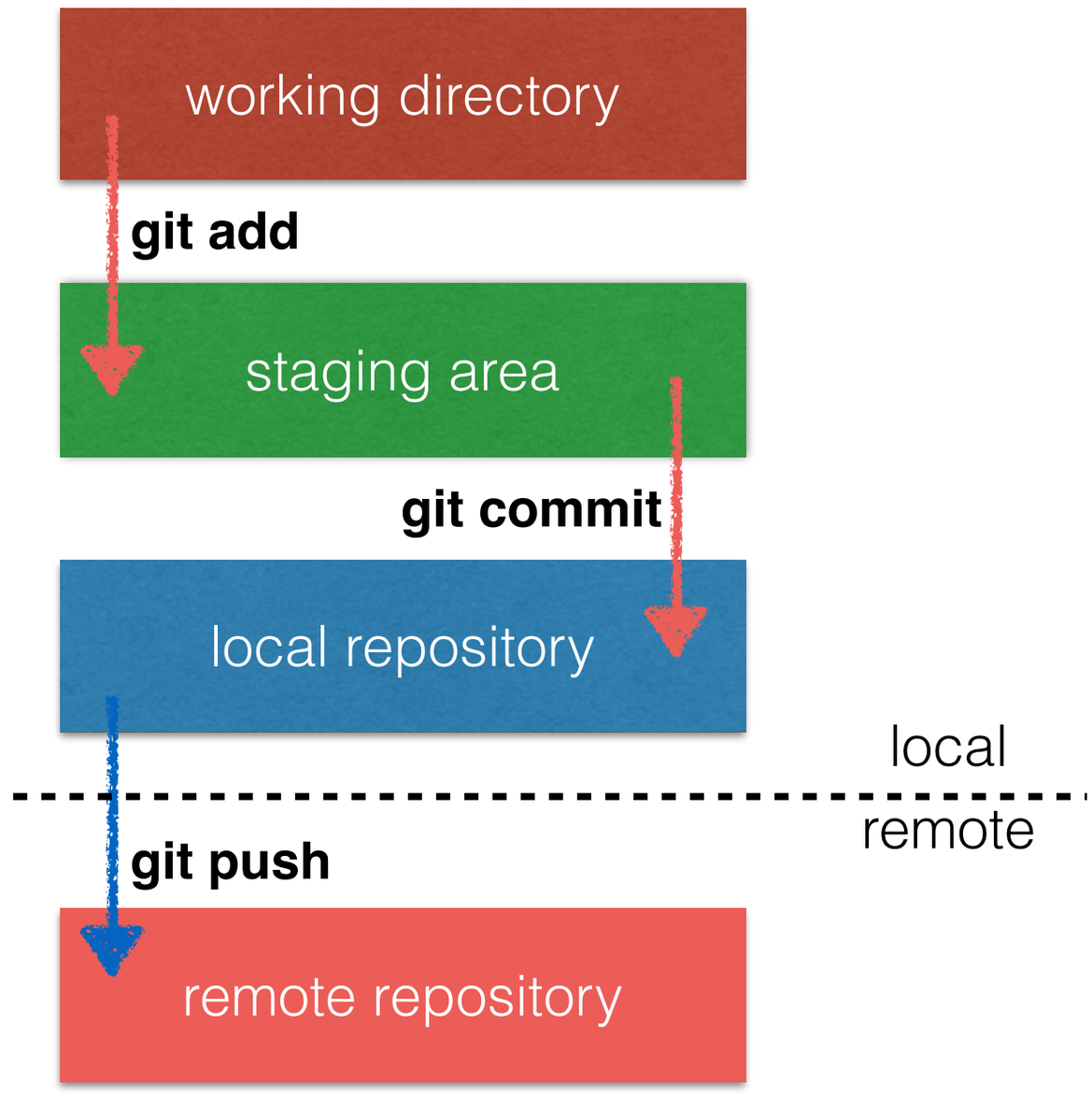
1. Create Git Repositories

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



2. Add/Commit/Push

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



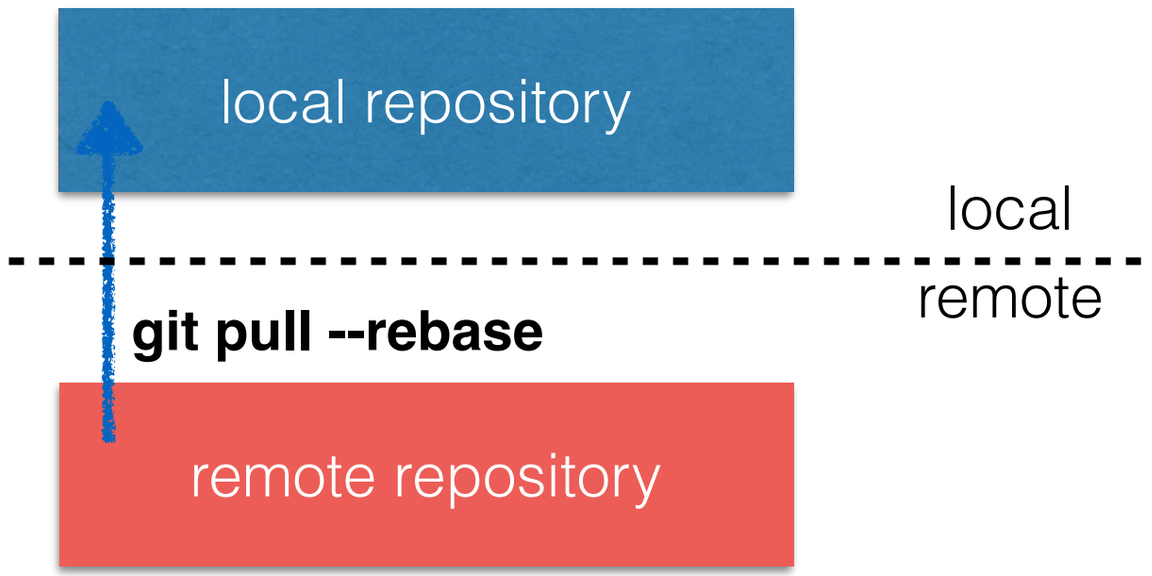
3. Check Status

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

```
|gh-pages ✗| → git status
On branch gh-pages
Changes to be committed:
  (use "git reset HEAD <file>..." to uns
        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 v                    ed)
  (use "git checkout -- <file>..." to                     working dire

        modified:   index.html
        modified:   init.md

Untracked files:
  (use "git add <file>..." to inclu                       committed)

        fork.md
```

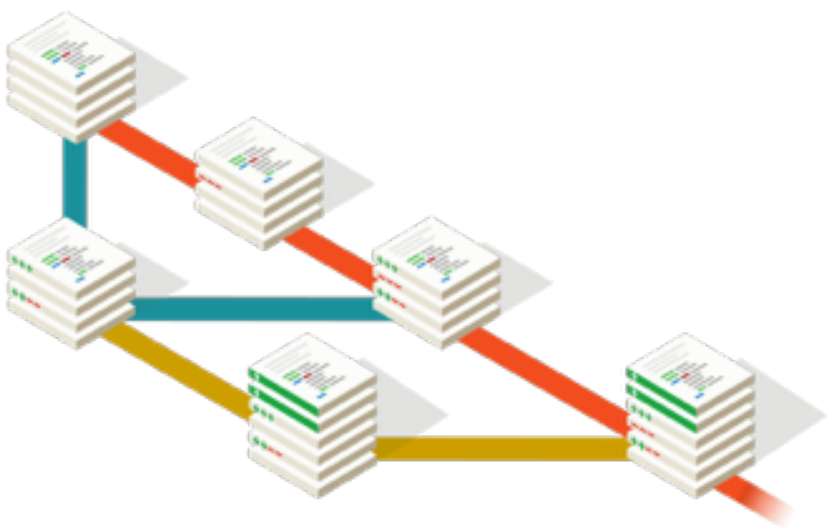
4. Sync Changes & Resolve Conflicts

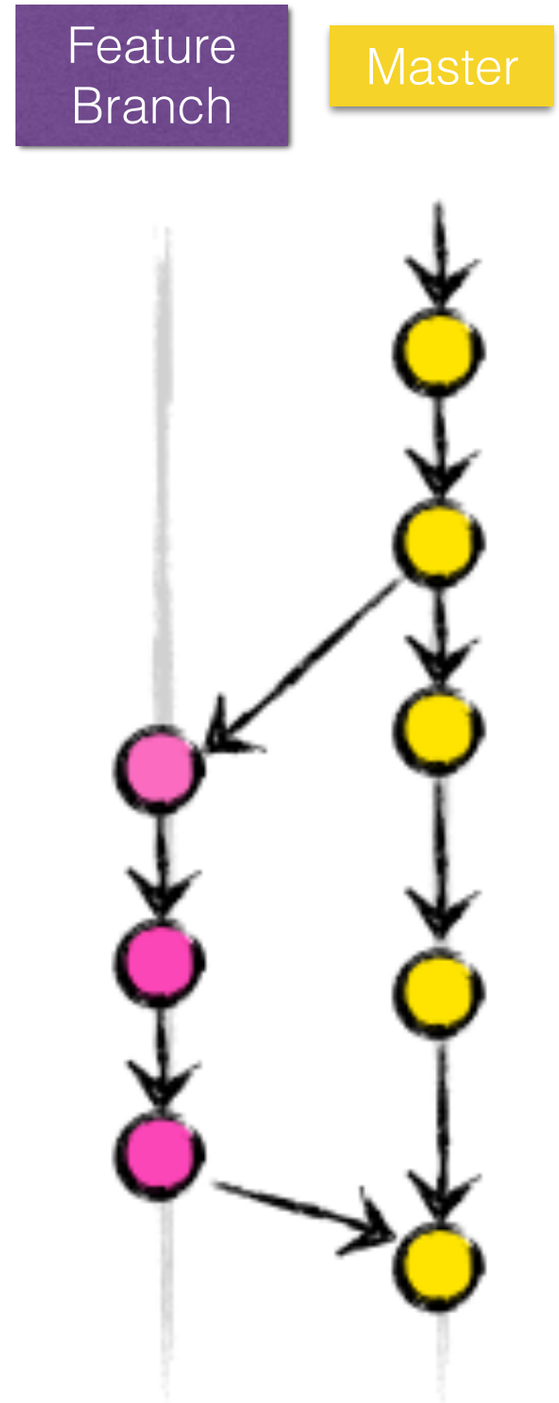
- **`git pull --rebase`** (referred approach)
- `git pull`
- `git status`
- `// clean up conflicts`
- `git add <conflicted file>`
- `git rebase --continue`
- `git push`



5. Branches

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



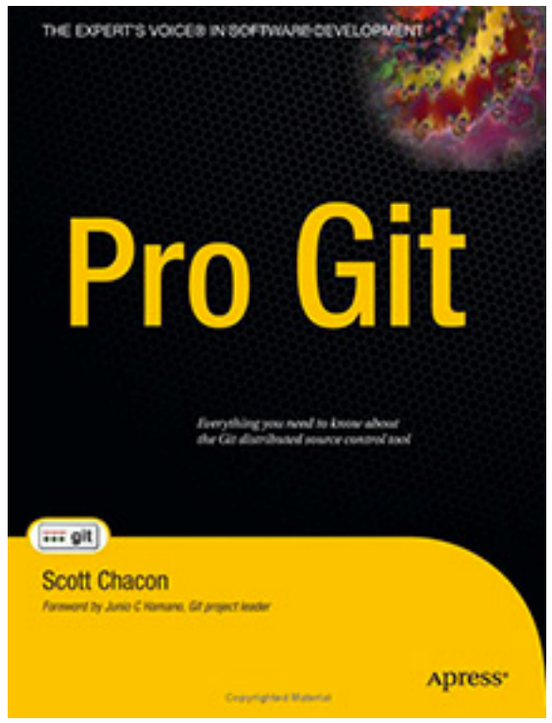


6. Git Undo

- Always backup first
- Google the solution

Git Learning Resources

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



Git Basics Overview