Intro to Git

How to Git Good

Redbrick
DCU's Networking Society

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.
COOL. HOW DO WE USE IT?
NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

What is Git

Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

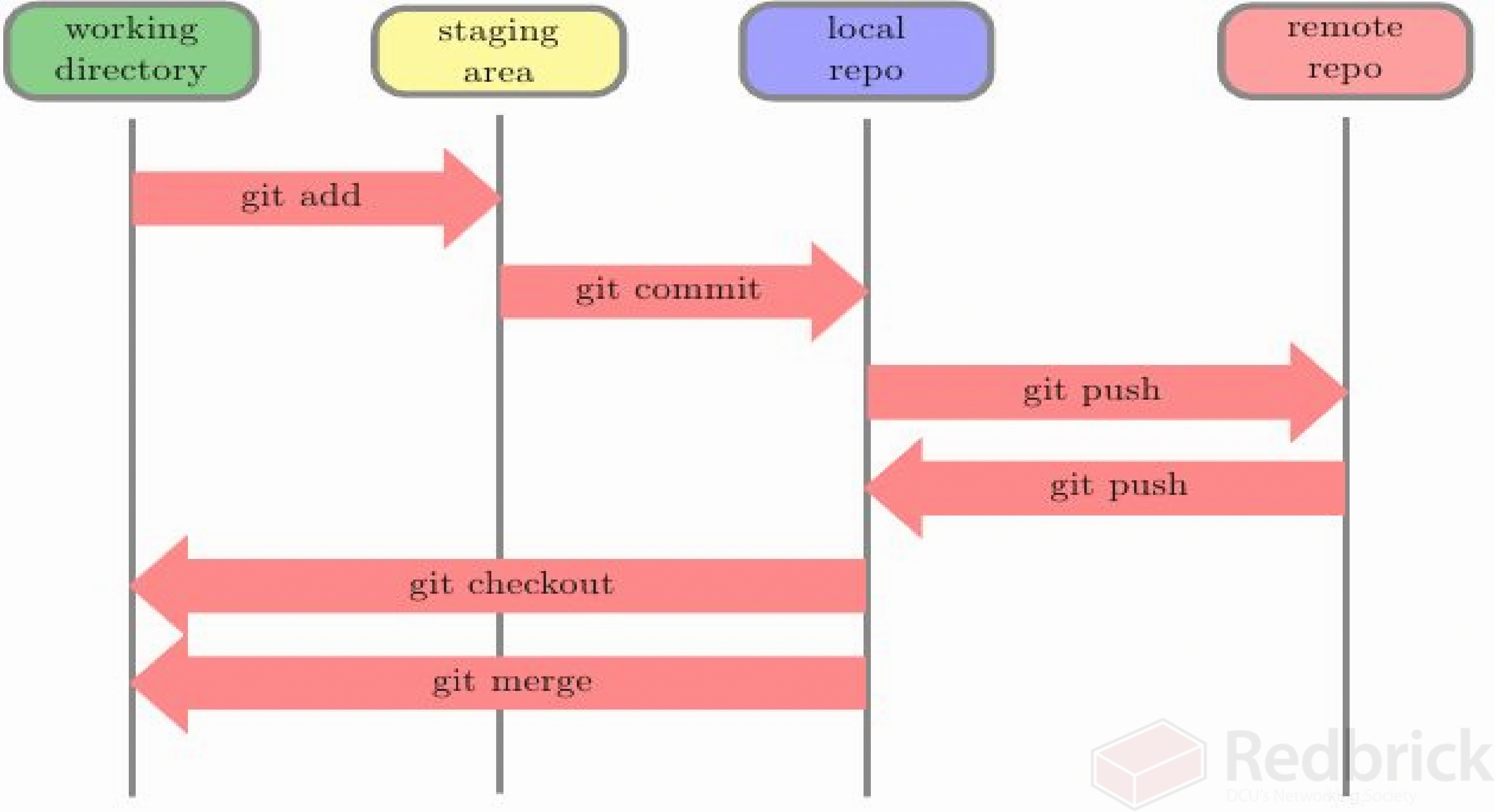
Redbrick
DCU's Networking Society

Why is Git good?

Git is scalable, has a small footprint and helps your project stay on target**

-- Linus Torvalds, creator of Git

*** this is not a quote*



working directory

staging area

local repo

remote repo

git add

git commit

git push

git push

git checkout

git merge

How to Git

- **Repo** - Is the place where all our code is stored
- **Fork**
  - Creates a new repo with all the same files in it
- **Clone**
  - cloning is copying the repo locally to work on it
  - *git clone http://github.com/redbrickdcu/introtogit.git*
- **Stage**
  - marks a file to go in your next commit
  - *git stage $fileYouChanged*
- **Commit**
  -
  - *git commit -m “Your commit message of what you changed”*
- **Push**
  - pushes your commits to the repo
  - *git push*
- **Pull**
  - pulls any new commits from the repo
  - *git pull*

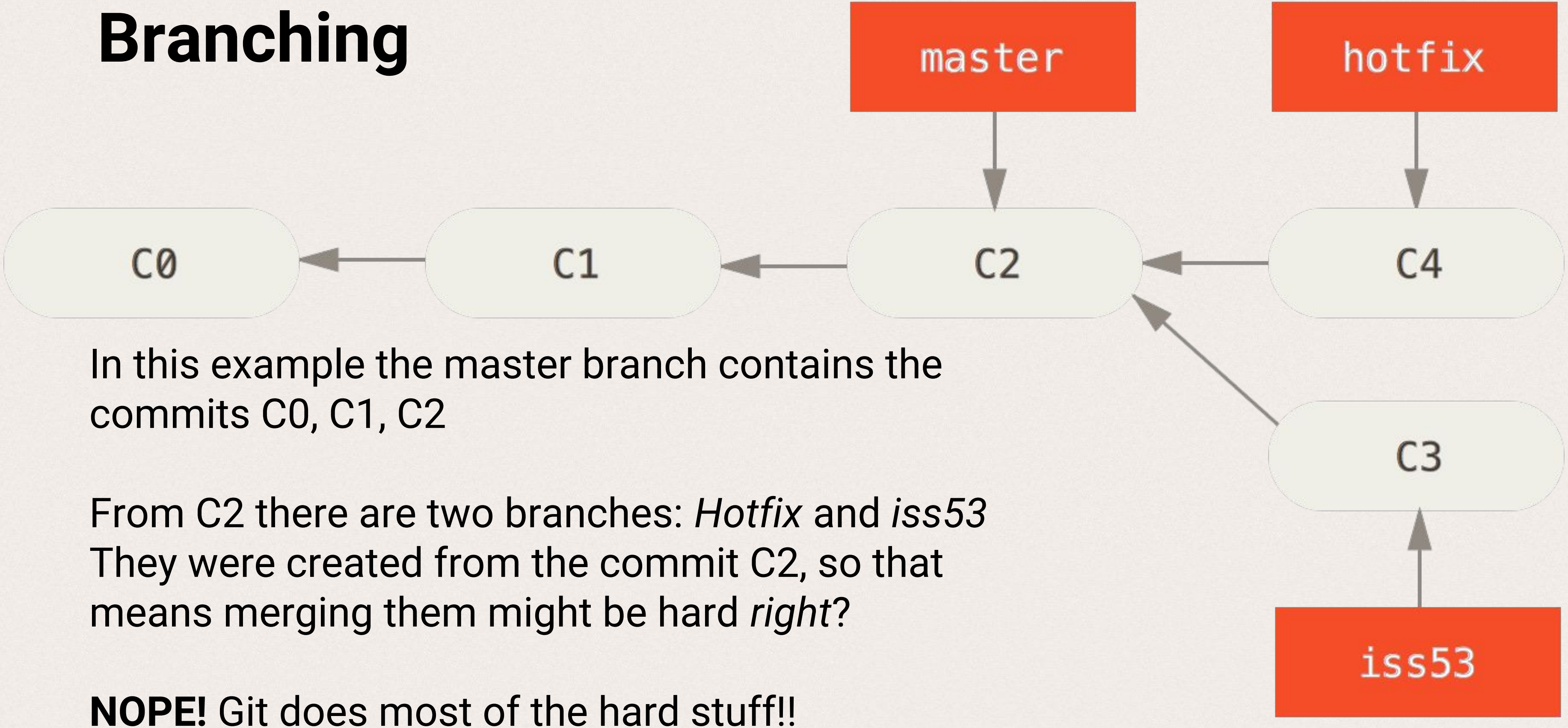
Redbrick
DCU's Networking Society

Git advanced

Branching

- Branches are essential to the git workflow
- Essentially a git project is made up of a series of changes to a directory, called commits.
- When you branch a repository you make a copy of those series of commits and then diverge. You can commit to this branch and it won’t affect the original stream

Branching



In this example the master branch contains the commits C0, C1, C2

From C2 there are two branches: *Hotfix* and *iss53*
They were created from the commit C2, so that means merging them might be hard *right*?

NOPE! Git does most of the hard stuff!!

Branching

- **Checkout**
  - Is a way to switch and create branches
  - To create a new branch
    - *git checkout -b $branchname*
  - To switch branch
    - *git checkout $branchname*
- **Merging**
  - Join two or more development histories together.
  - First checkout the branch you want to merge into, then run:
    - *git merge $otherbranch*

Github

- Github.com
- Public & Private repos
- Issues
  - Labels
- Easily fork
- Easy to create Pull requests and get involved

Workshop

github.com/redbrick/introToGit

Demo Fail

What to do when it breaks

Question