

# CS 520

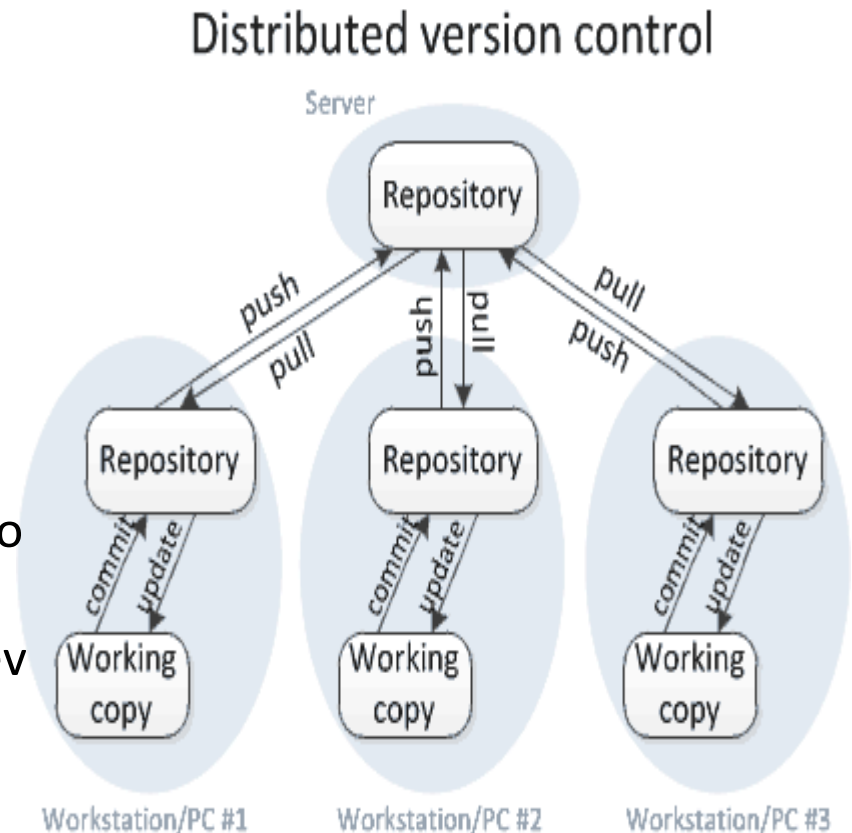
Theory and Practice of Software  
Engineering  
Spring 2021

**In-class exercise 1: Advanced uses of Git**

February 25, 2021

# Types of VCS -- Distributed

- Each developer has their own repository.
  - Created by the developer, or
  - Cloned from an existing (*remote*) repository
- Developers work on their own repos
  - They can commit, branch, etc.
  - Activity is local unless it is *pushed* to remote repo
  - Remote activity is not seen until dev *fetches* from the remote repo
- Examples: Mercurial (hg), git



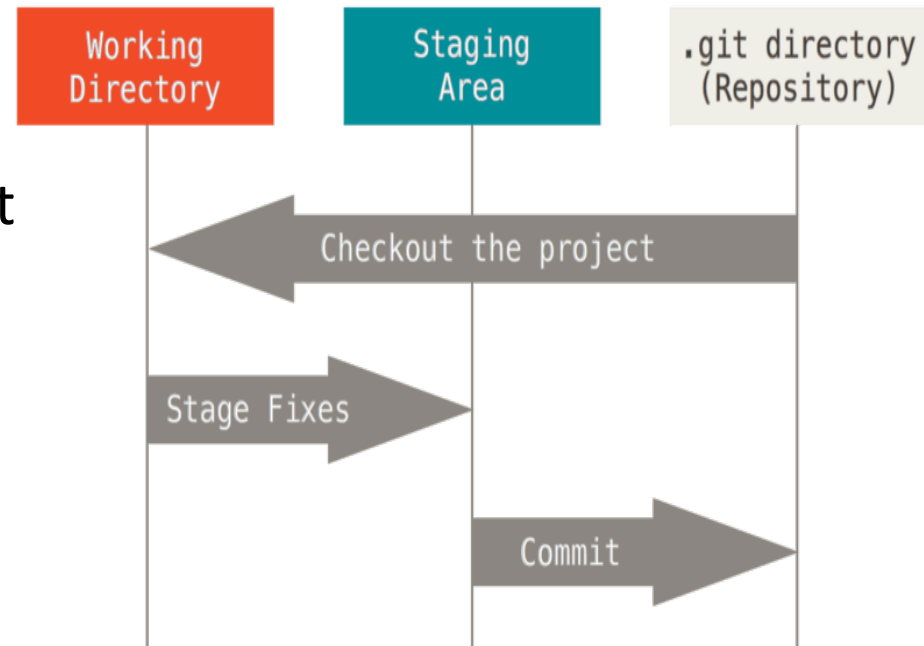


**git**

<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>  
<https://github.com/>

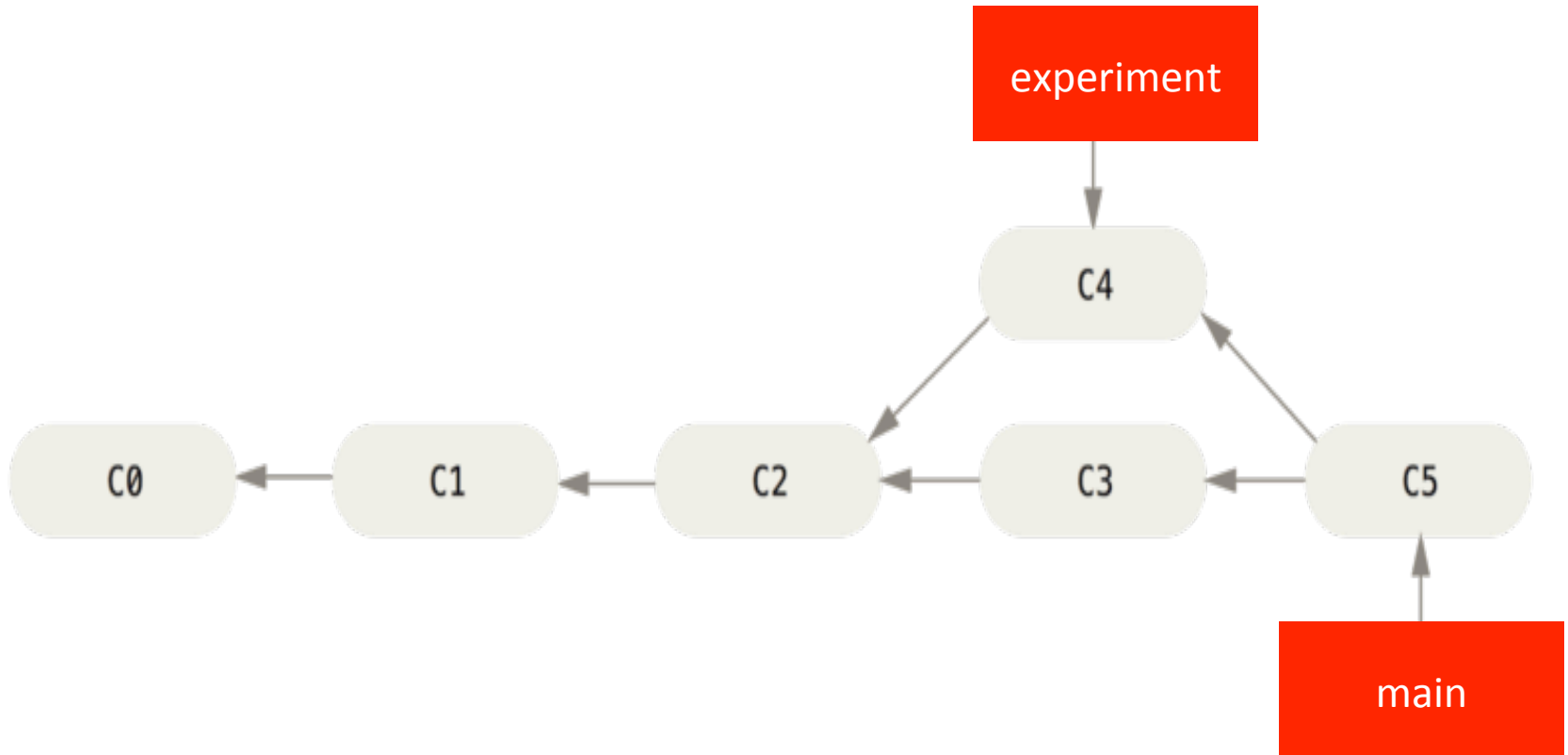
# Git -- Three Main Stages

1. **Committed:** Everything in the file is currently in the database
2. **Modified:** Changed the file but have not committed to the database
3. **Staged:** Marked the file for addition to the database in the next commit



Note that all of the above pertain to *tracked* files.

# Git -- Git repo abstraction



# Git -- Git Commit Object

Message

Parent

Hash

basic functionality with sanity check done

🔗 master (#15)

👤 Rico Angell committed on Oct 2, 2017

1 parent 1c82903 commit 2c2ec0a72ce21615dae3868af32d929852807a65

Author or committer along with date

📄 Showing 2 changed files with 108 additions and 64 deletions.

Unified Split

Initially entire file then diffs

29 Themis2.0/test/test\_themis.py

View

@@ -1,3 +1,4 @@

```
1 import pytest
2 import themis
3
```

```
1 + from itertools import chain, combinations
2 import pytest
3 import themis
4
```

@@ -26,23 +27,23 @@ def test\_get\_test\_result():

```
26
27 def test_group_discrimination():
28     t = themis.Themis(xml_fname="settings.xml")
29 - _, p = t.group_discrimination(i_fields=["Sex", "Race"])
30 - print "Sex and Race: ", p
31 -
32 - _, p = t.group_discrimination(i_fields=["Race"])
33 - print "Race: ", p
34 -
35 - _, p = t.group_discrimination(i_fields=["Sex"])
36 - print "Sex: ", p
```

```
27
28 def test_group_discrimination():
29     t = themis.Themis(xml_fname="settings.xml")
30 + print "\nGroup:"
31 + for f in t._all_relevant_subs(["Sex", "Race", "Age", "Income"]):
32 +     _, p = t.group_discrimination(i_fields=f)
33 +     print f, "--> ", p
```

```
37
38 def test_causal_discrimination():
39     t = themis.Themis(xml_fname="settings.xml")
40 - _, p = t.causal_discrimination(i_fields=["Sex", "Race"])
```

```
34
35 def test_causal_discrimination():
36     t = themis.Themis(xml_fname="settings.xml")
37 + print "\nCausal:"
```

# Today's in-class exercise

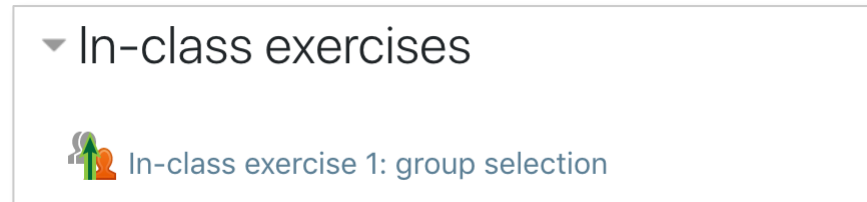
- Instructor: Heather Conboy
- Grader: Sayantan Bhowmik

NOTE) May also get help from git man pages, lecture materials, and other on-line resources

<https://people.cs.umass.edu/~hconboy/class/2021Spring/CS520/in-class1.pdf>

# Group selection

- Form 4- or 5-person teams
  - If you need more members in your team, raise your hand and ask the instructor
- Use Moodle to self-select a team



- Select a team member responsible for submitting the completed exercise by next Wednesday



# Set up

1. Check if you are using a newer version of git (version  $\geq$  2.7.4)
2. Clone the basic-stats git repository:  
git clone <https://github.com/LASER-UMASS/basic-stats> basic-stats
3. Create a (second) local fork by locally cloning **again**, this time from the first local clone:  
git clone basic-stats basic-stats-fork

*NOTE) Recommend taking notes as you go for writing up the answers to the six (6) questions as well as capturing some of the git logs along the way*