

Version Control with Git

- **Before we start**
 - **Sign up at github.com**
 - **Partner up with the person next to you**

What is Version Control?

(AKA revision control, source control)

- Tracks changes to files
- Any file can be tracked
- Text (.txt, .csv, .py, .c, .r etc.) works best
 - These allow smart *diff / merge* etc.

Why Use Version Control? #1

- A more efficient backup
- Reproducibility



Why Use Version Control? #2

- Teamwork



Version Control Tracks Changes



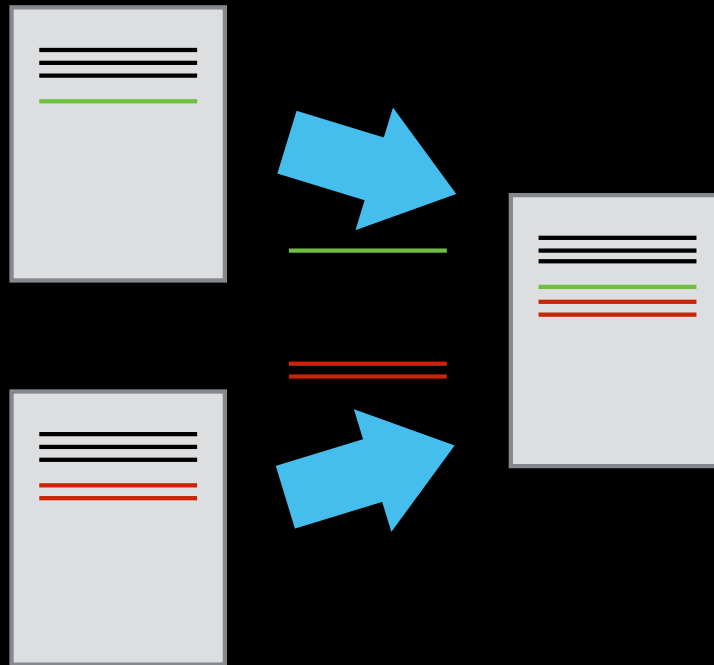
- **Changes are tracked sequentially**

Version Control Tracks Changes



- Different versions can be saved

Version Control Tracks Changes



- Multiple versions can be merged

Version Control Alternatives

- **Subversion (svn) - Centralised**
- **Mercurial (hg) - Distributed**
- **Git (git) – Distributed**

- **N.B. GitHub != git**

Local Configuration

- **git config**

Getting Demo Files

- **git clone**
<https://github.com/Southampton-RSG/2019-11-19-southampton-swc>

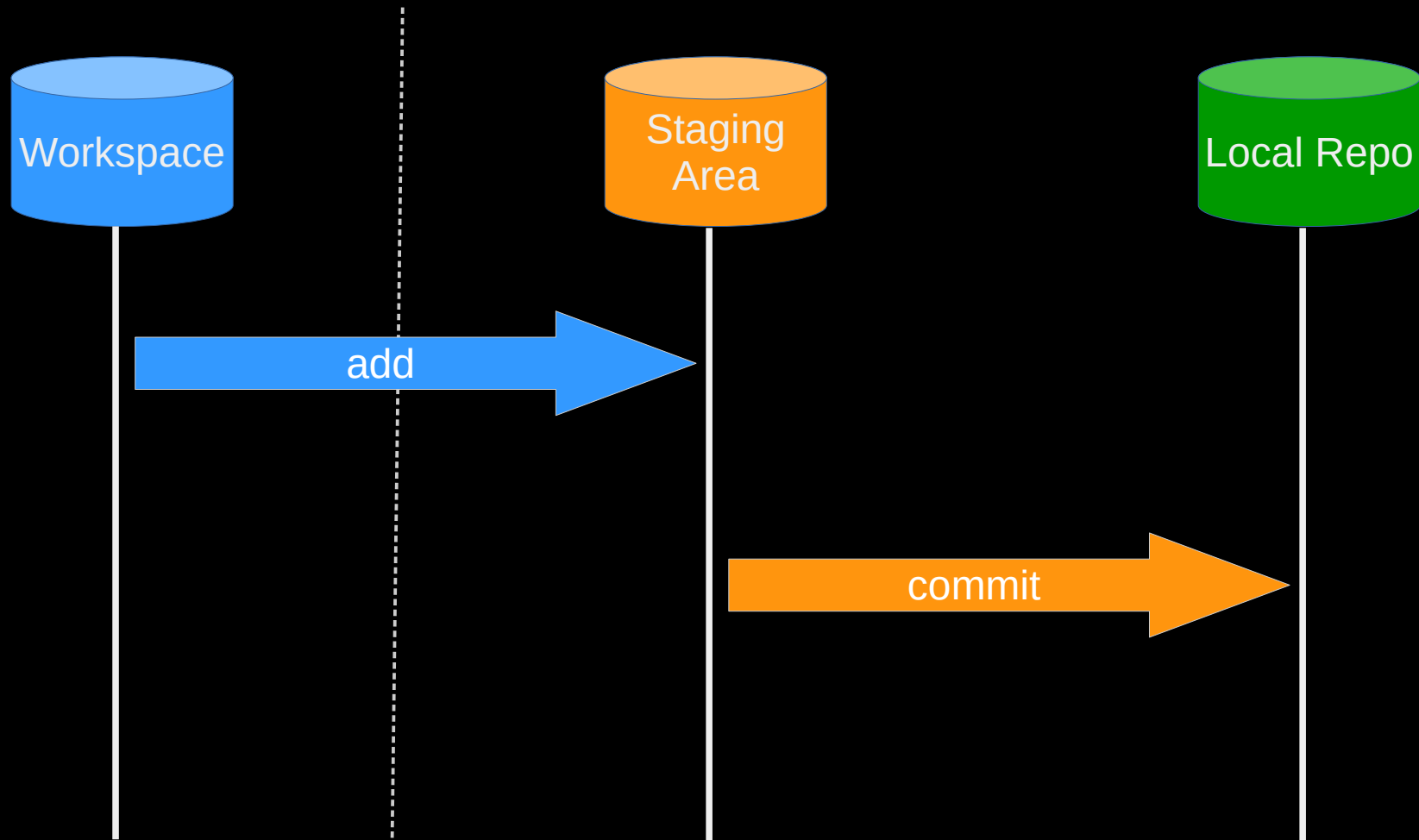
Creating a Repository

- **git init**
- **git status**

Tracking Changes to Files

- **git add**
- **git commit**

Git – add and commit



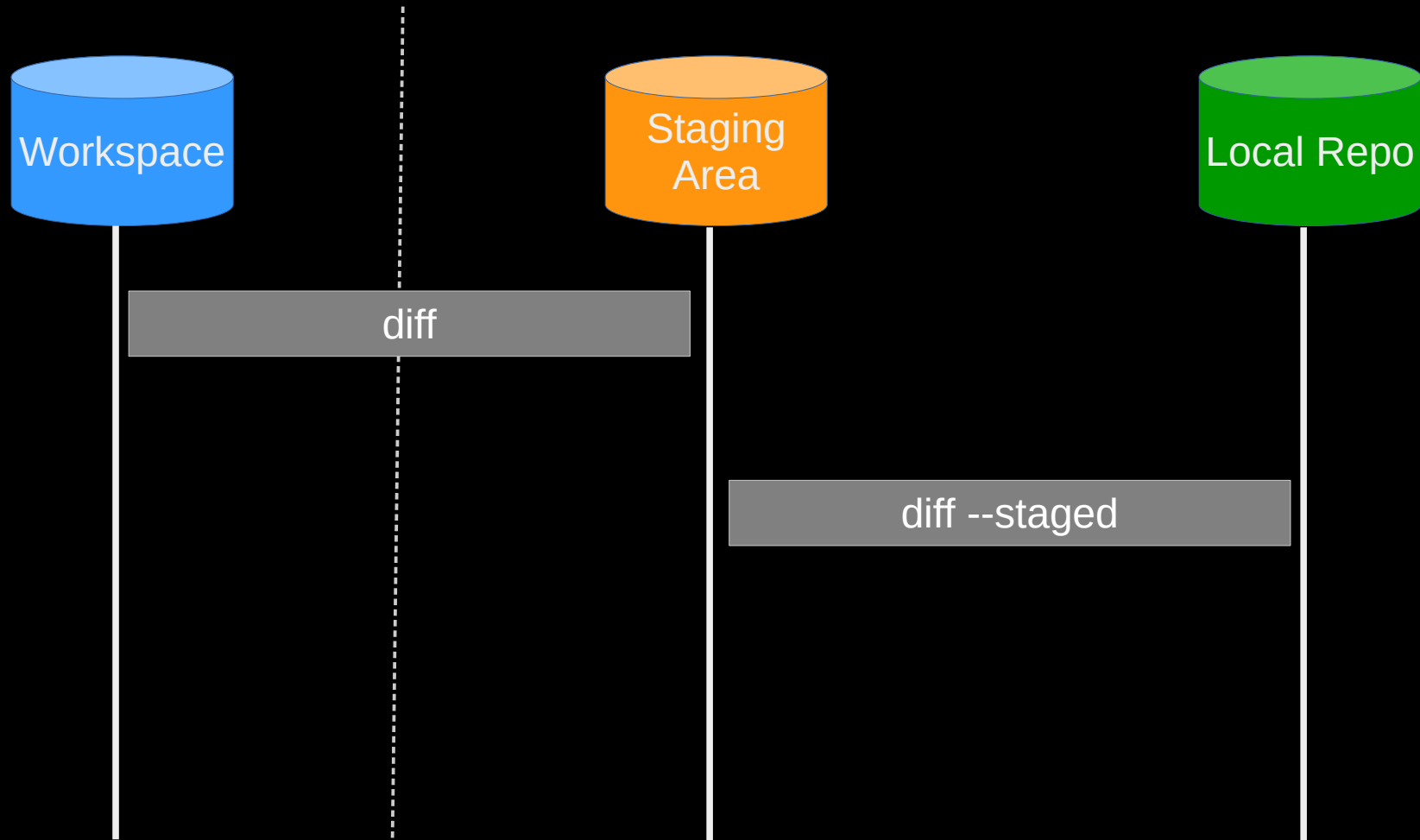
Visible File System

Git Repository

Exploring History #1

- **git log**
- **git diff**

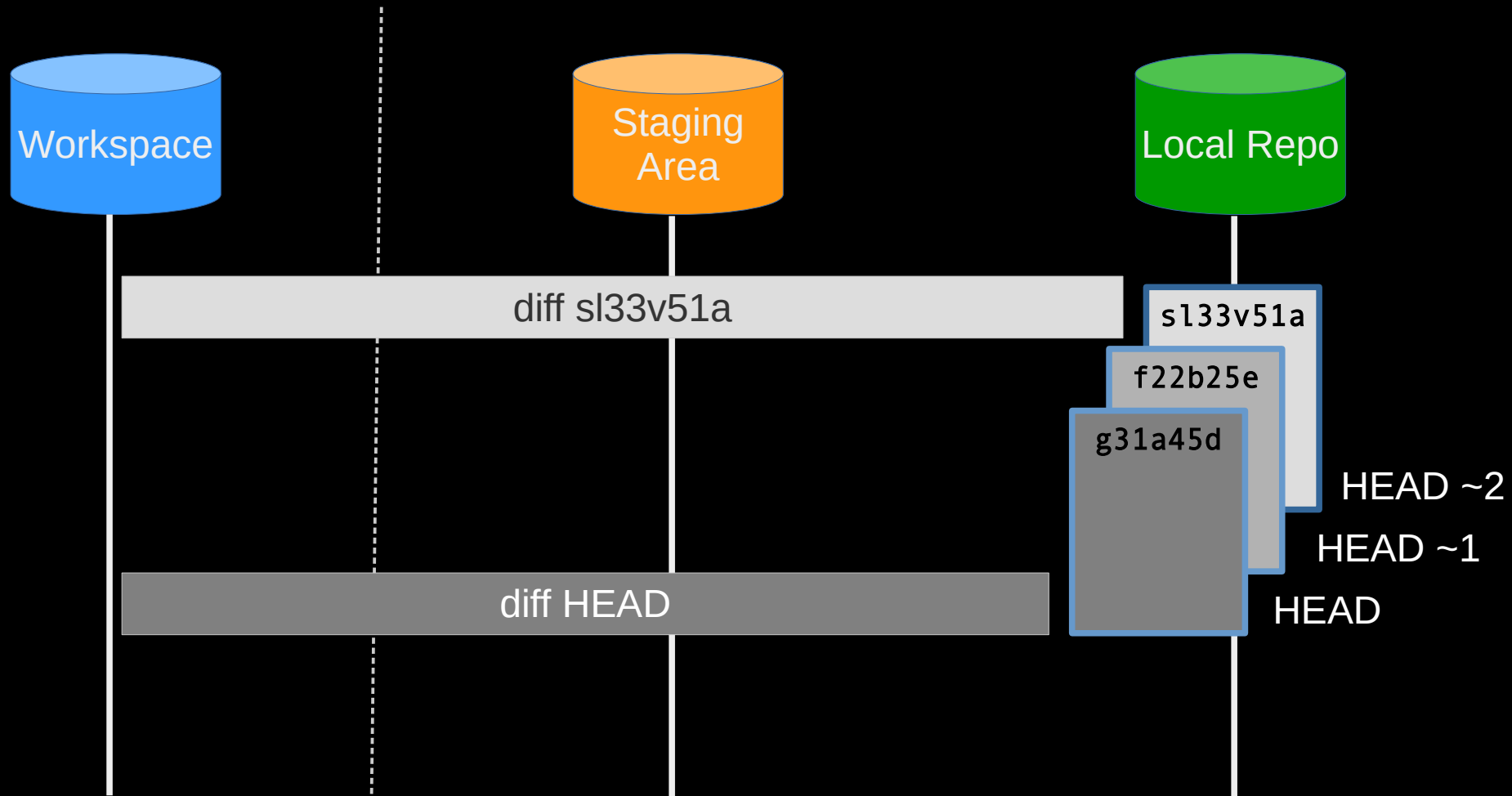
Git - diff #1



Visible File System

Git Repository

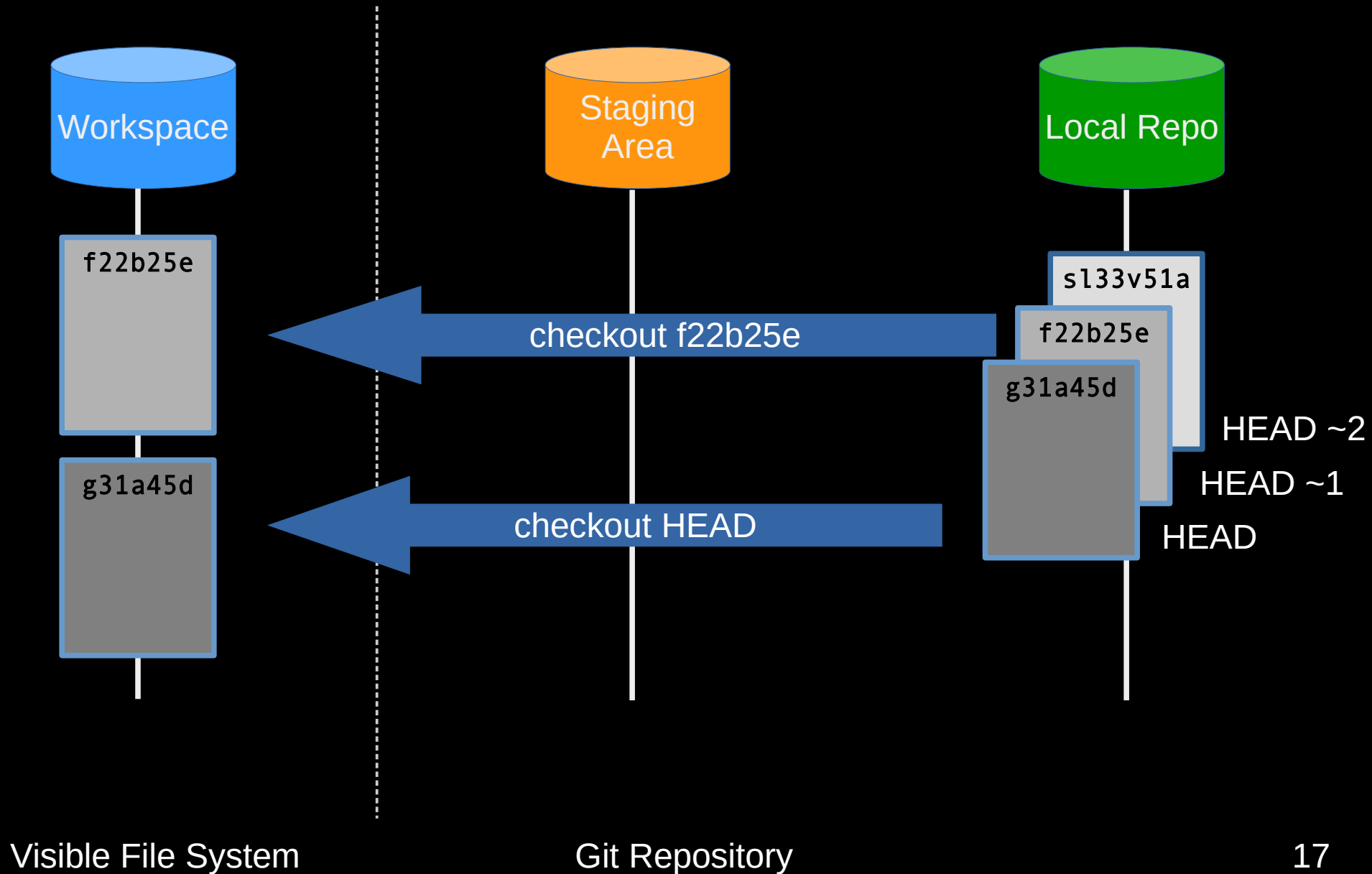
Git - diff #2



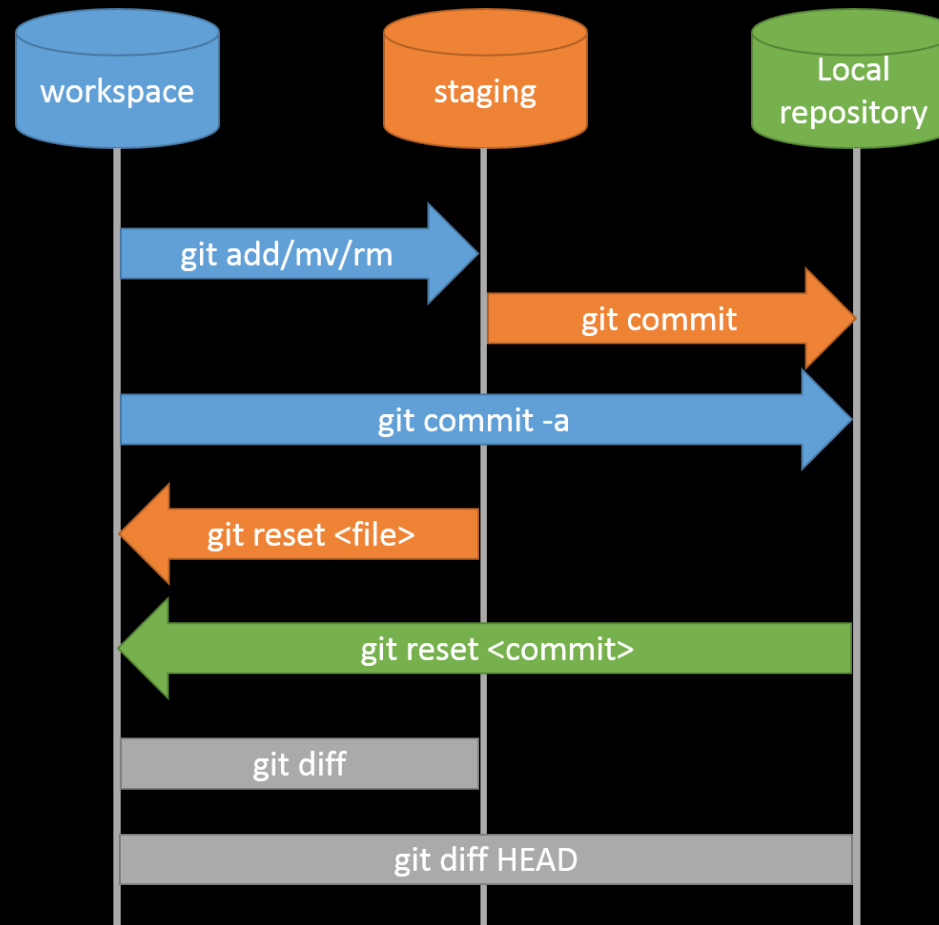
Restoring Files

- **git checkout**

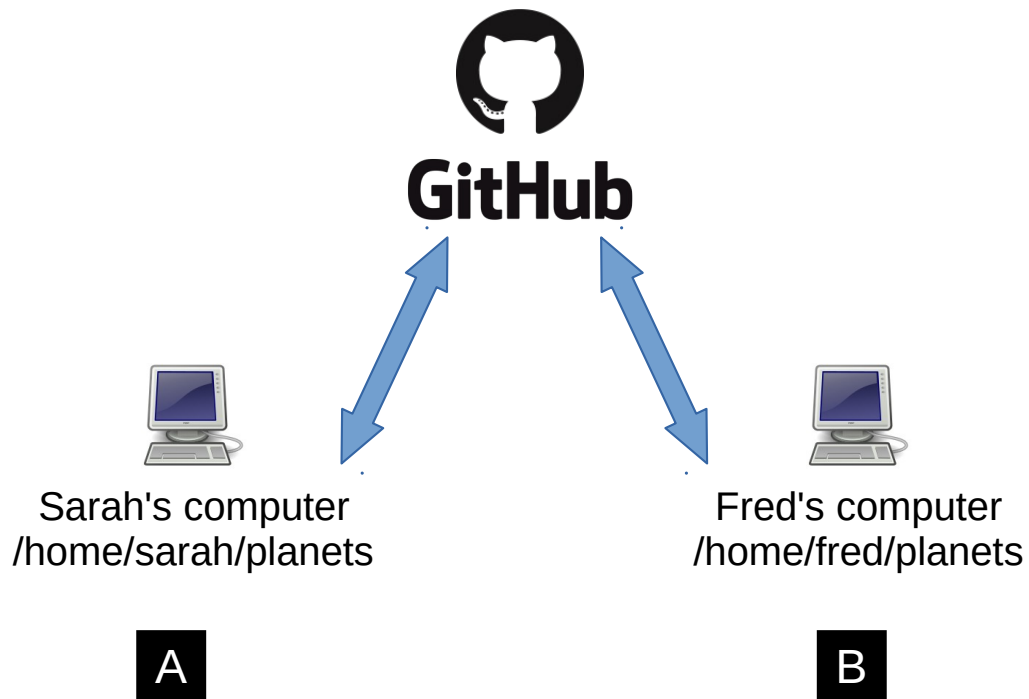
Git - restoration



Git Workflow – Local Repo.



Collaboration



Collaboration

- **Let's collaborate via remote repo**
- **Working in pairs:**
- **Developer A / Developer B**

Collaboration:

Remote Repositories #1

- Developer A
 - Sign in <https://github.com/>
 - Create repository
 - git remote add
 - git push
 - Add Developer B as a collaborator

Collaboration:

Remote Repositories #2

- Developer B
 - Clean up
 - git clone
 - git add
 - git commit
 - git push

Collaboration:

Remote Repositories #3

- Developer A
 - `git pull`

Collaboration:

Remote Repositories #4

- Exercise
 - Developer A – Add README.md, (authors and info).
 - Developer B – sync up your repository

Collaboration: Conflicts #1

```
mm = inches * 25.4  
return mm
```



```
mm = inches * 25.4  
return mm
```

```
# TODO(Sarah): Add  
fu
```

Collaboration: Conflicts #2

```
mm = inches * 25.4  
return mm
```

```
graph TD; A["mm = inches * 25.4  
return mm"] --> B["mm = inches * 25.4  
return mm  
# TODO(Sarah): Add fu"]; A --> C["mm = inches * 25.4  
return mm  
# TODO(Fred): Add fu"]; B --> D["?"]; C --> D;
```

```
mm = inches * 25.4  
return mm
```

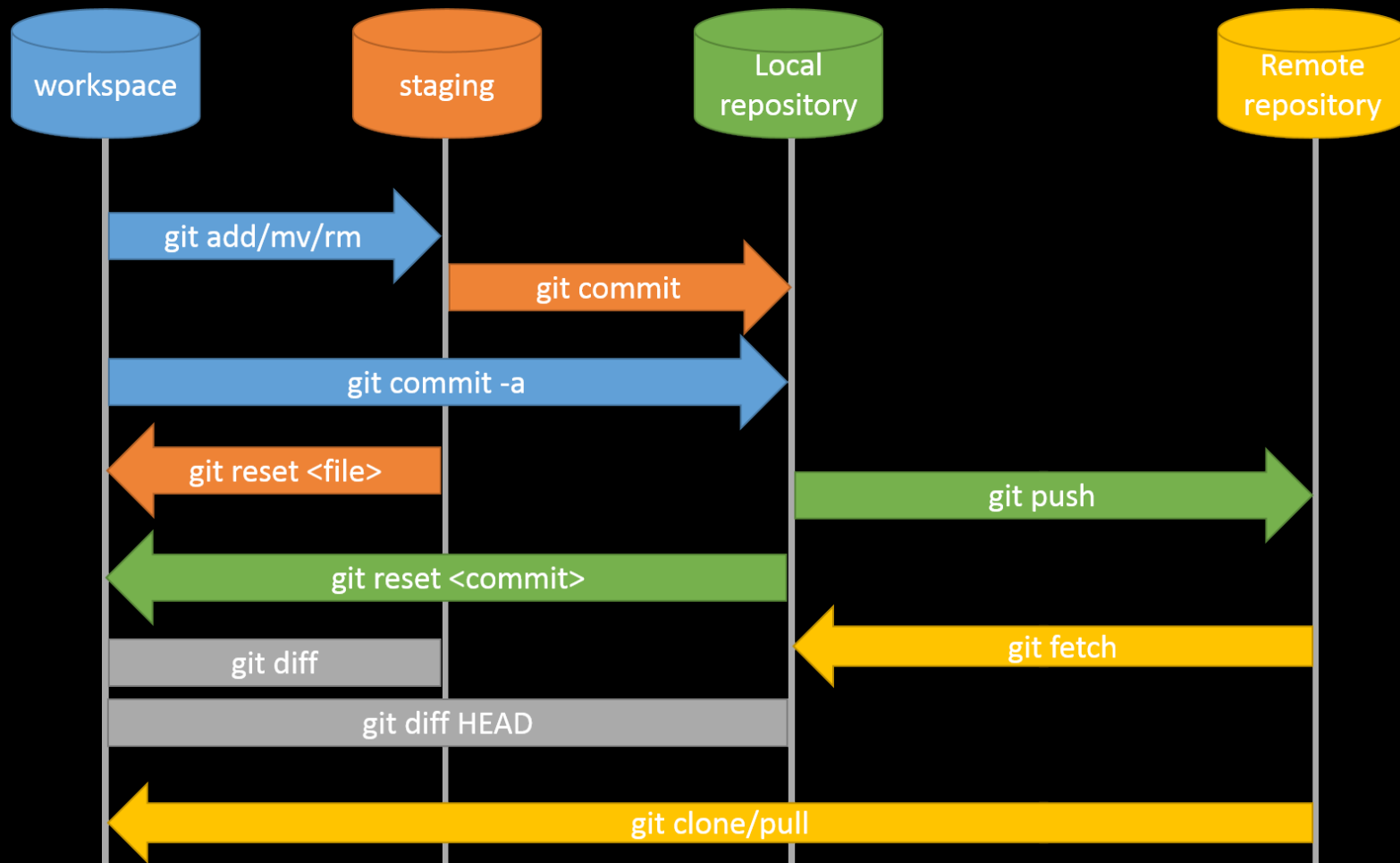
```
# TODO(Sarah): Add  
fu
```

```
mm = inches * 25.4  
return mm
```

```
# TODO(Fred): Add fu
```

?

Git Workflow – Remote Repo.



What next?

- **Branching / Merging**
- **<https://software-carpentry.org>**