

Practical Computing for Scientists

Armin Sobhani
CSCI 2000U
UOIT – Fall 2015

High Speed Recap

High Speed Recap – Setup Git

\$ _

High Speed Recap – Setup Git

```
$ git config --global user.name "Your Name"
```

```
$ _
```

High Speed Recap – Setup Git

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "Your Email"  
$ _
```

High Speed Recap – Setup Git

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "Your Email"  
$ git config --global core.editor /usr/bin/nano  
$ _
```

High Speed Recap – Setup Git

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "Your Email"  
$ git config --global core.editor /usr/bin/nano  
$ git config --global color.ui auto  
$ _
```

High Speed Recap – Setup Git

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "Your Email"  
$ git config --global core.editor /usr/bin/nano  
$ git config --global color.ui auto  
$_
```

you only need to do these once...

High Speed Recap – Repository

\$ _

High Speed Recap – Repository

```
$ git init
```

```
$ _
```

High Speed Recap – Repository

```
$ git init  
$ git status  
$ _
```

High Speed Recap – Repository

```
$ git init
```

```
$ git status
```

```
$ git add file1 file2 ...
```

```
$ _
```

High Speed Recap – Repository

```
$ git init  
$ git status  
$ git add file1 file2 ...  
$ git commit -m "initial commit"  
$ _
```

High Speed Recap – Repository

```
$ git init  
$ git status  
$ git add file1 file2 ...  
$ git commit -m "initial commit"  
$ git log  
$ _
```

Online Resource



Online Resource



Check the *Git in 15 min*

Blackboard > Course Content > Week 4 (Oct. 5-9)
> Monday Oct. 5 > Git in 15 min

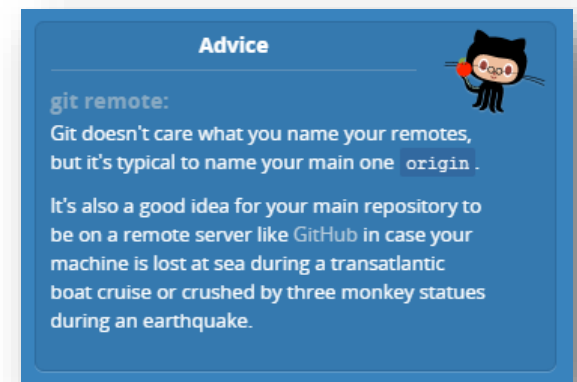
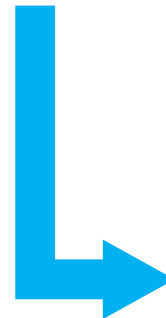
Online Resource



Check the *Git in 15 min*

Blackboard > Course Content > Week 4 (Oct. 5-9)
> Monday Oct. 5 > Git in 15 min

Remember to check Advice Box before running each command



Online Resource

Check the *Git in 15 min*

Blackboard > Course Content > Week 4 (Oct. 5-9)
> Monday Oct. 5 > Git in 15 min

Remember to check Advice Box before running each command

Continue till 1.10 Remote Repositories (about 6 min)

The Life Cycle of the File Status

The Life Cycle of the File Status

Untracked

The Life Cycle of the File Status

Untracked

Tracked

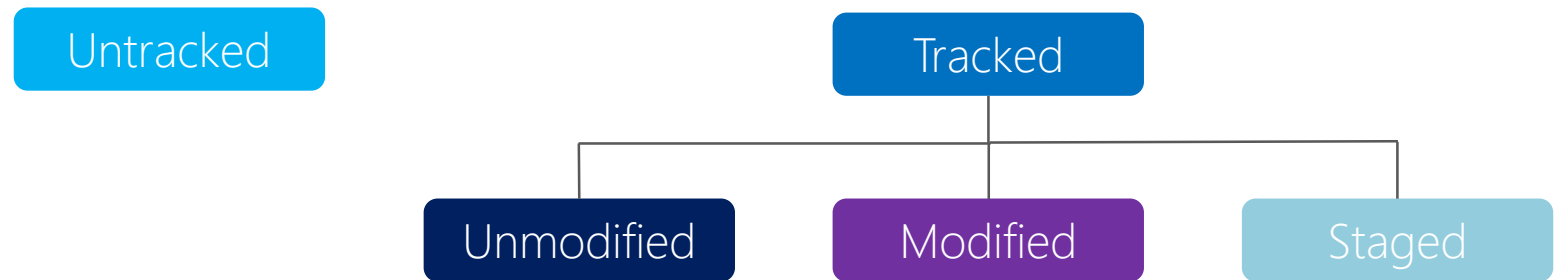
The Life Cycle of the File Status



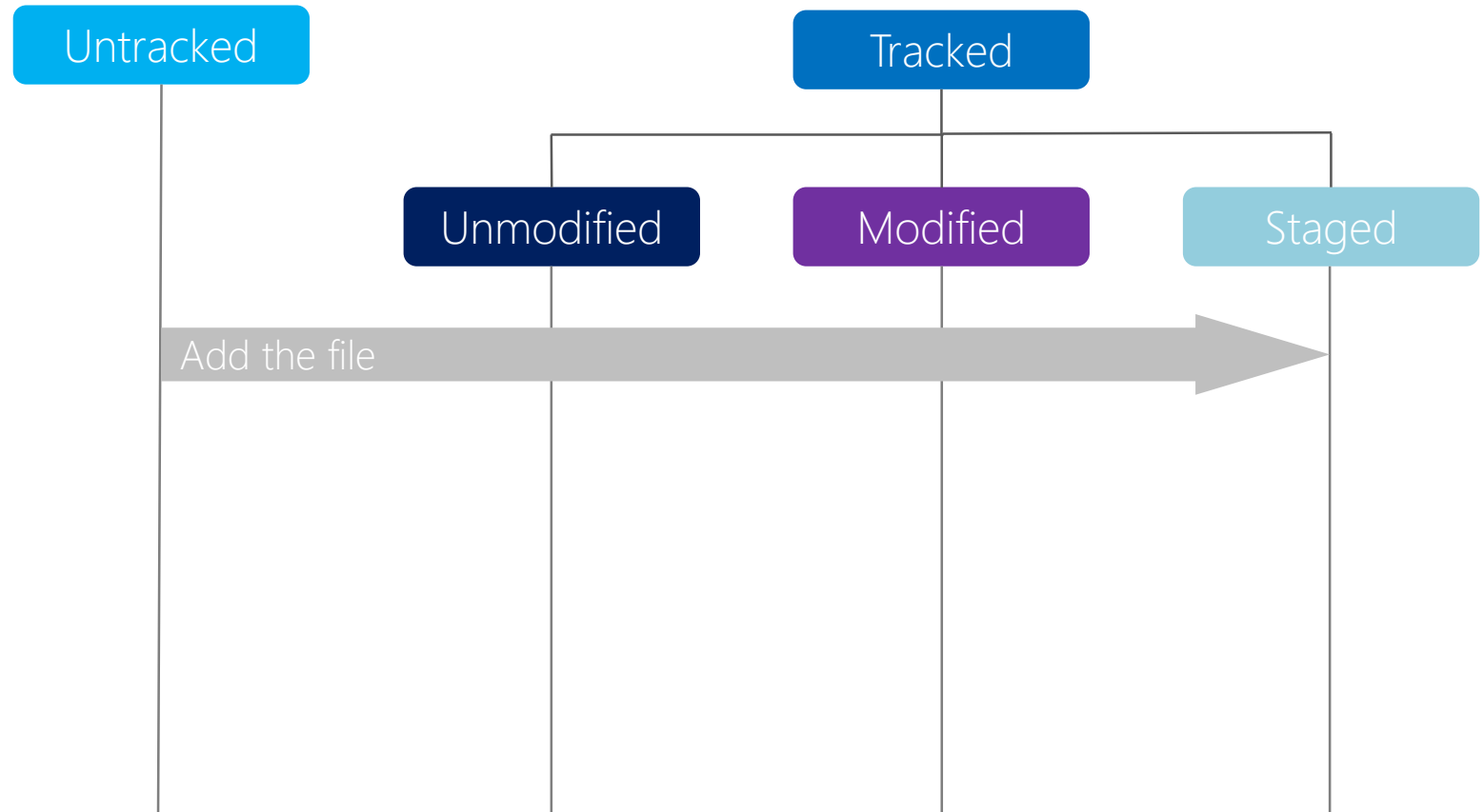
The Life Cycle of the File Status



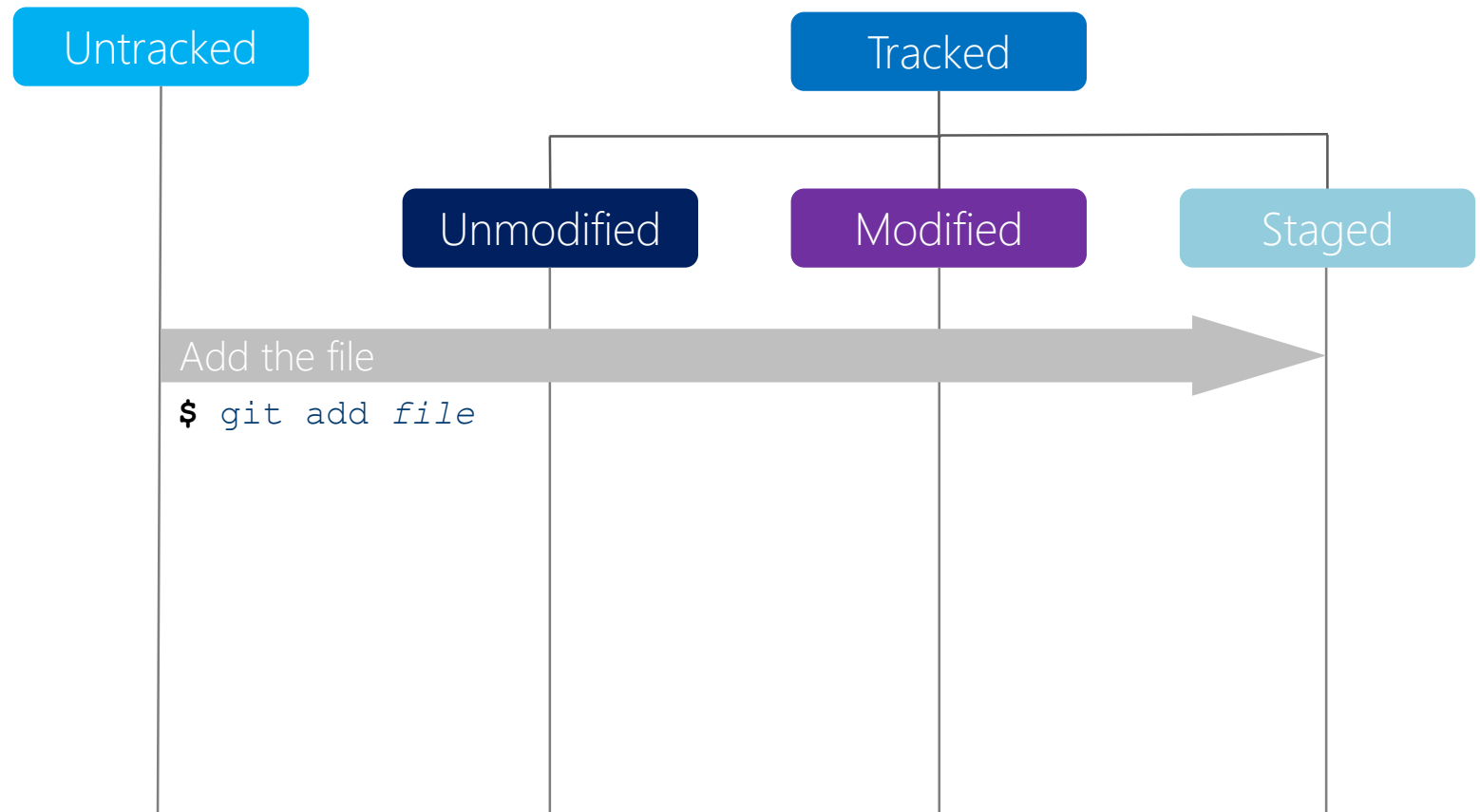
The Life Cycle of the File Status



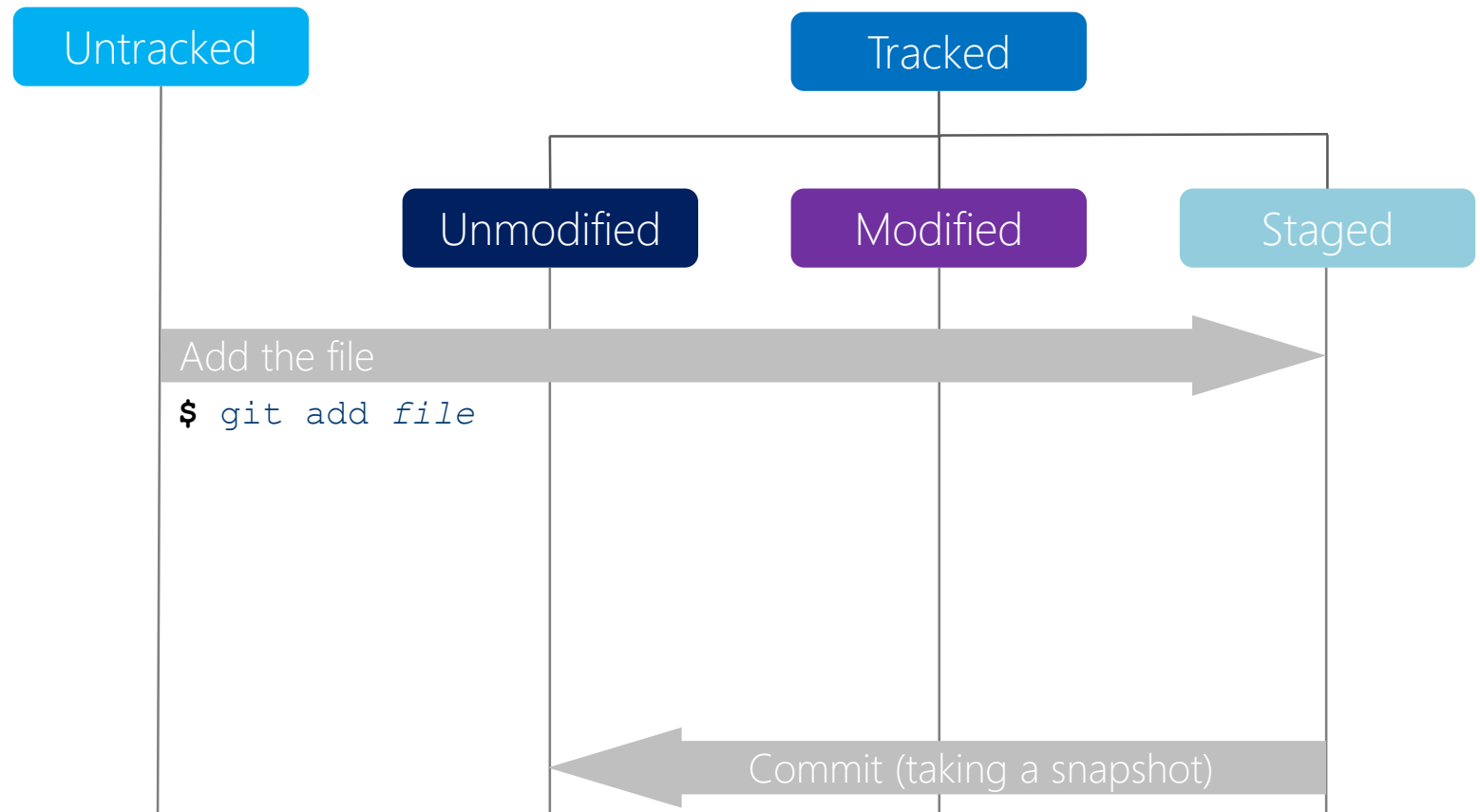
The Life Cycle of the File Status



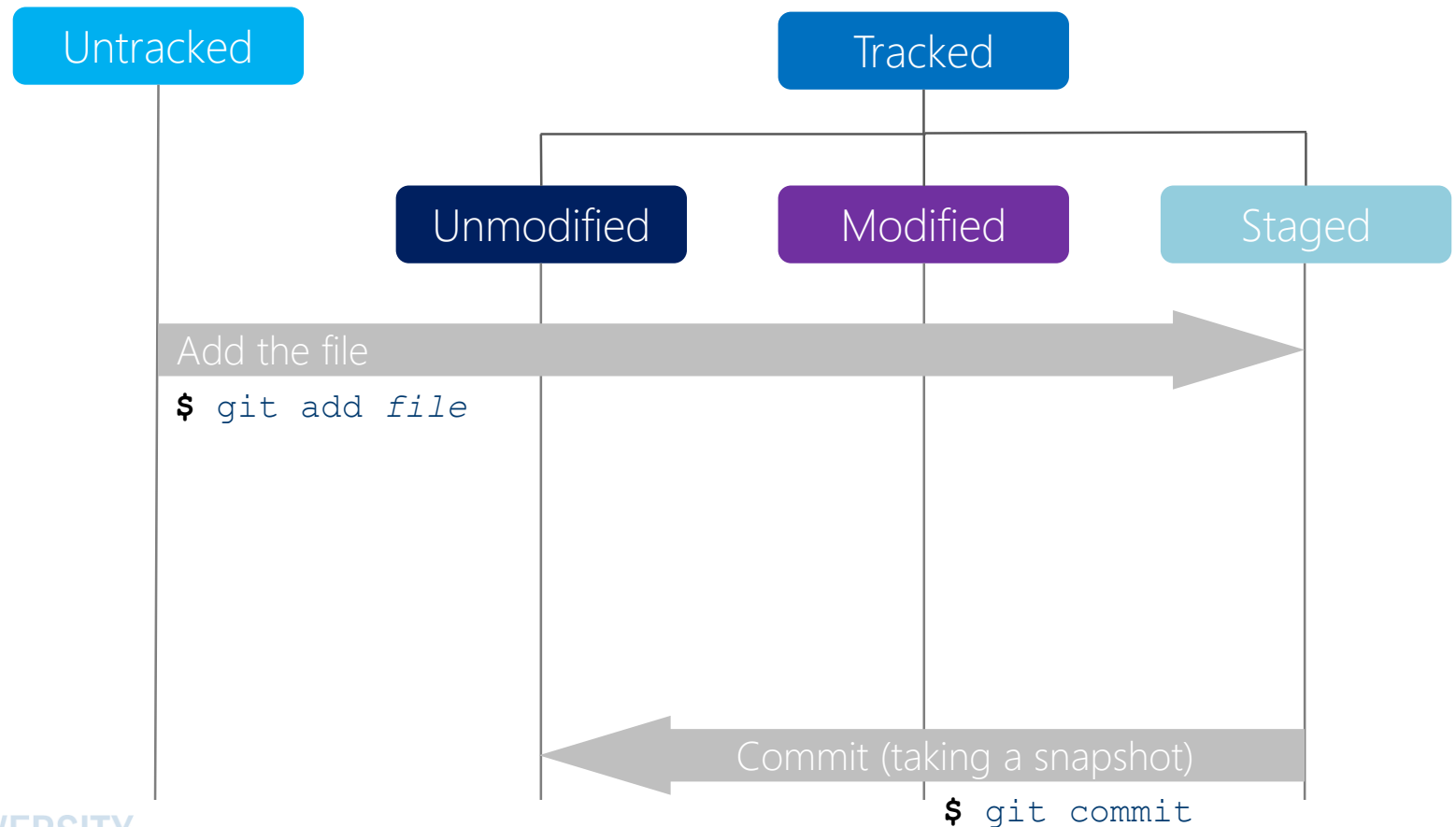
The Life Cycle of the File Status



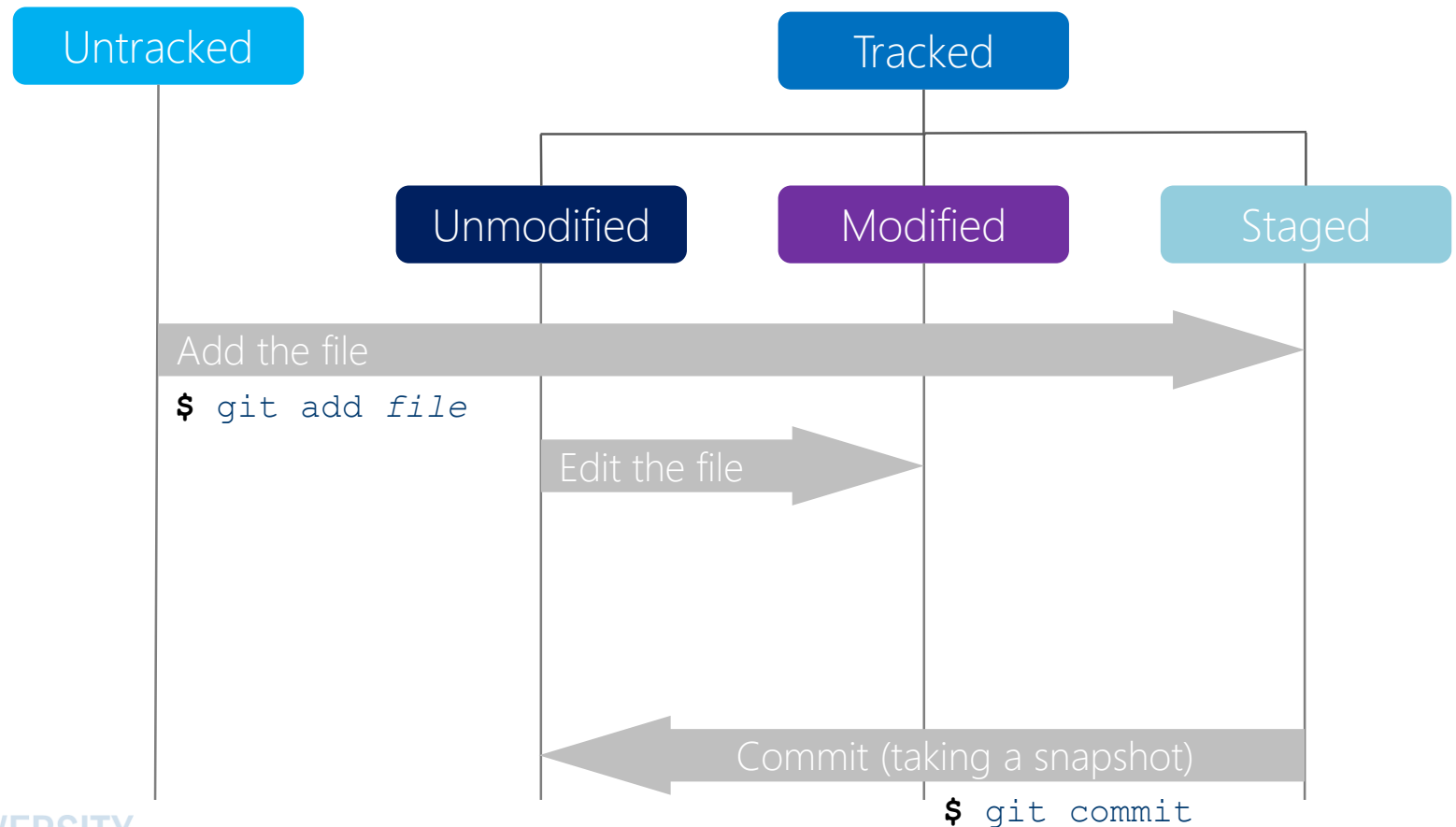
The Life Cycle of the File Status



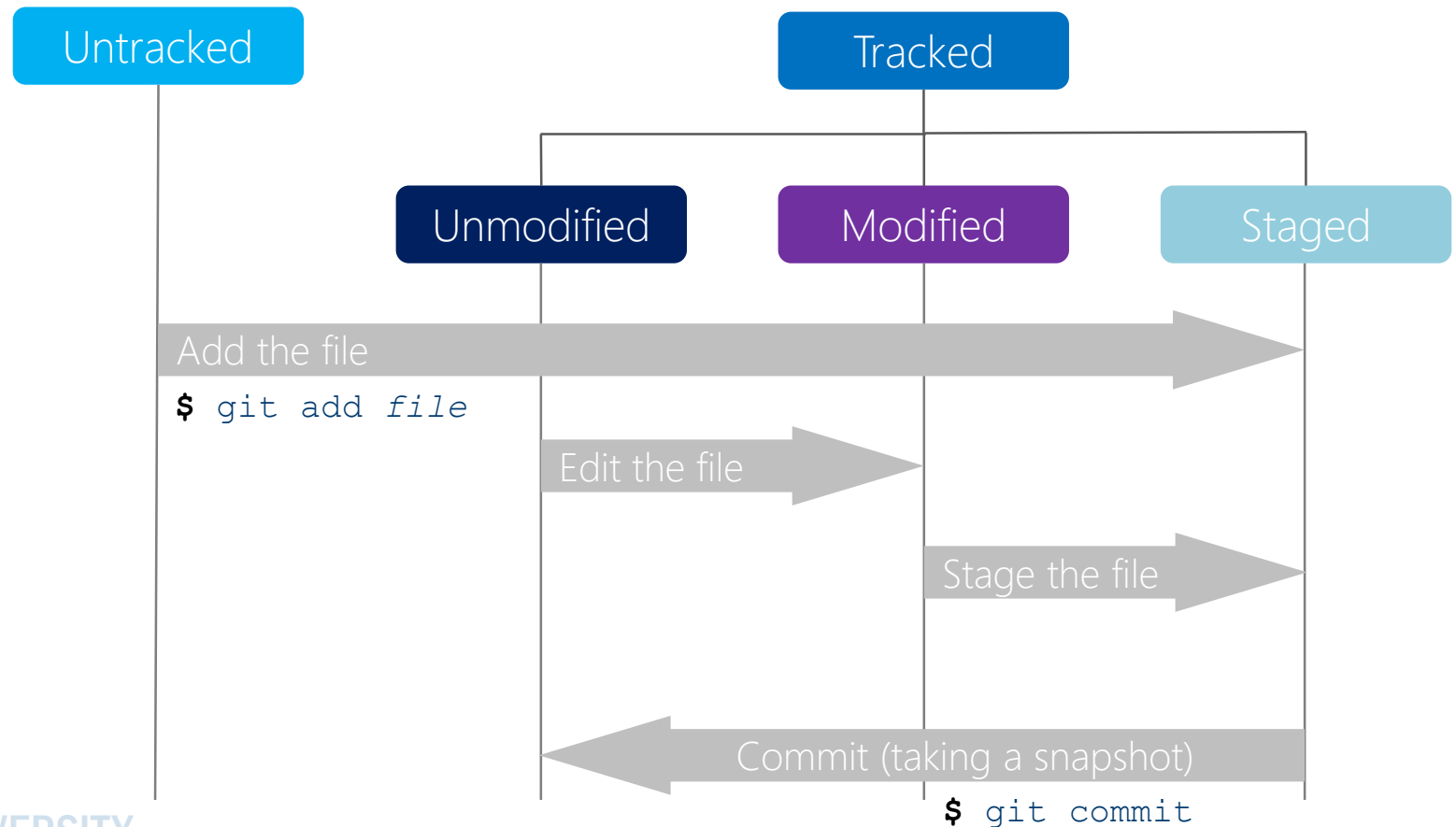
The Life Cycle of the File Status



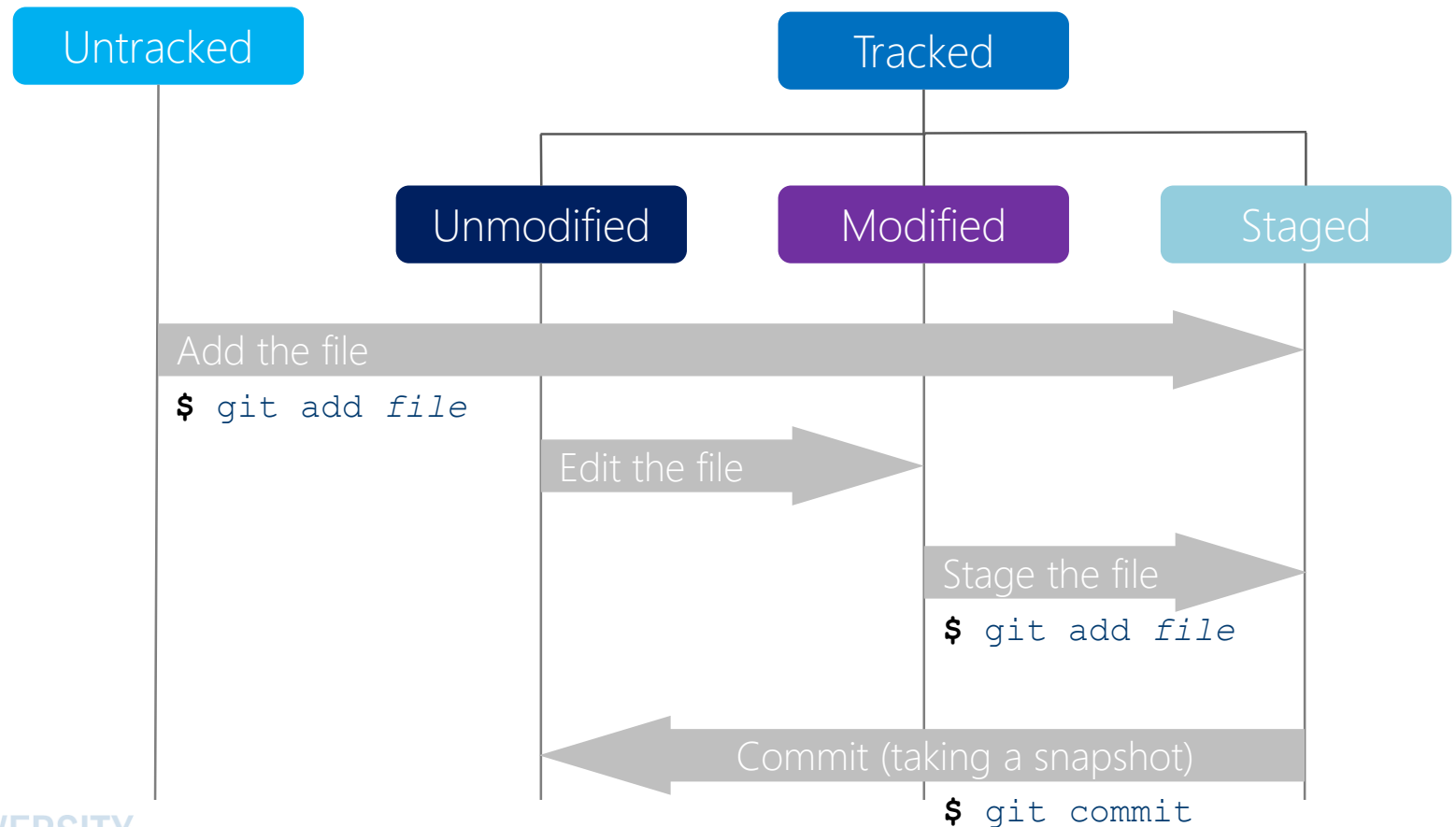
The Life Cycle of the File Status



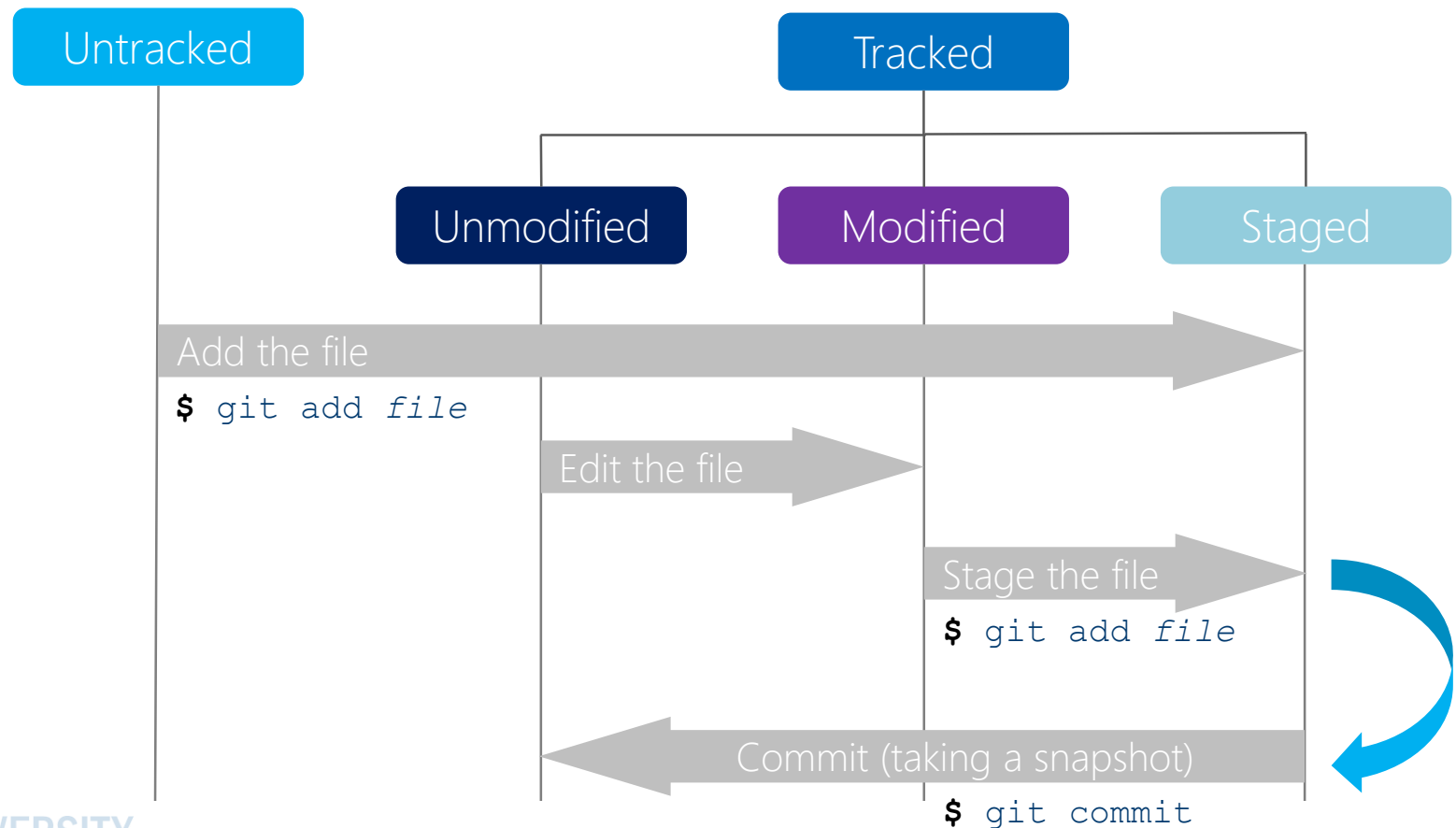
The Life Cycle of the File Status



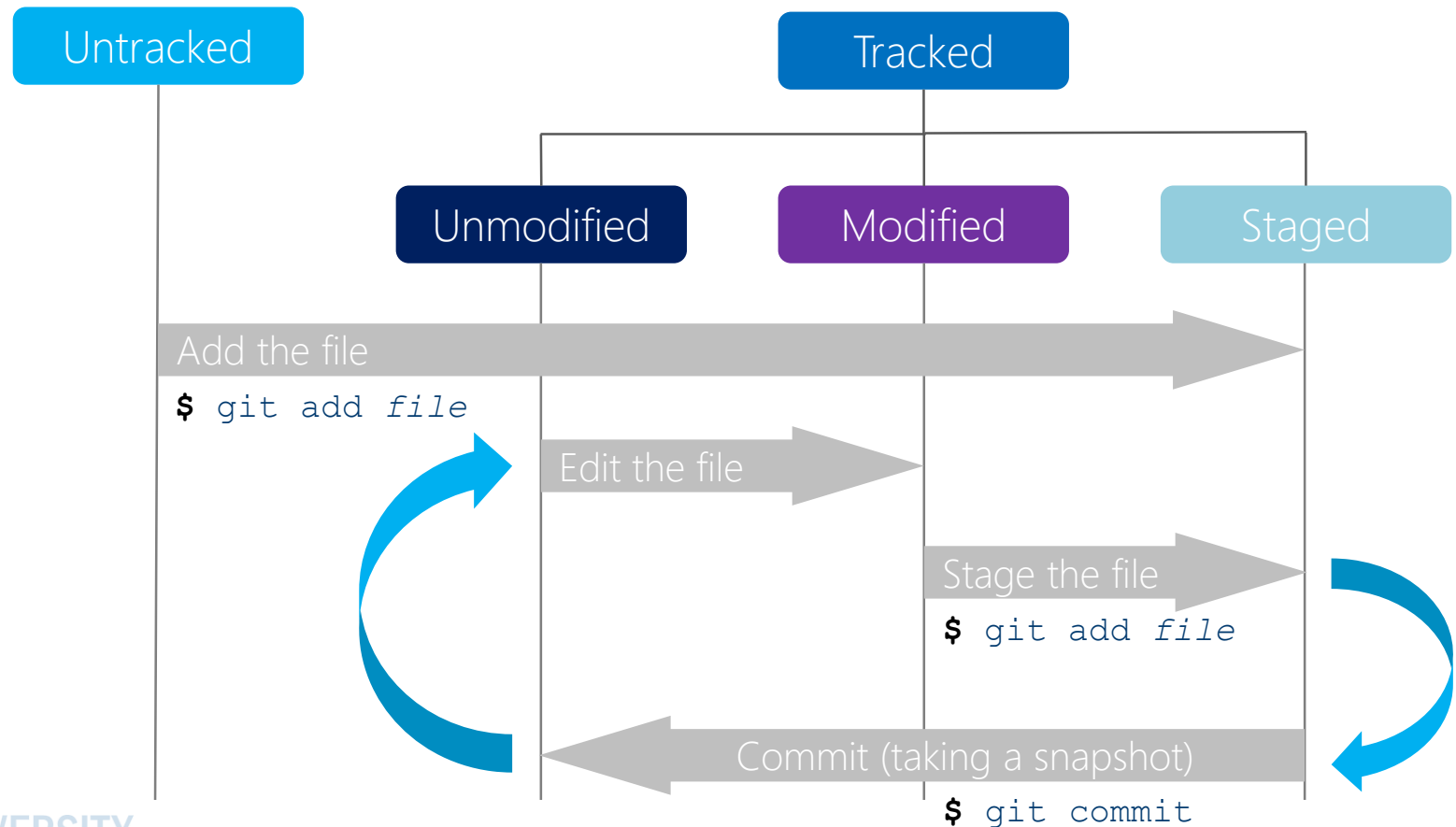
The Life Cycle of the File Status



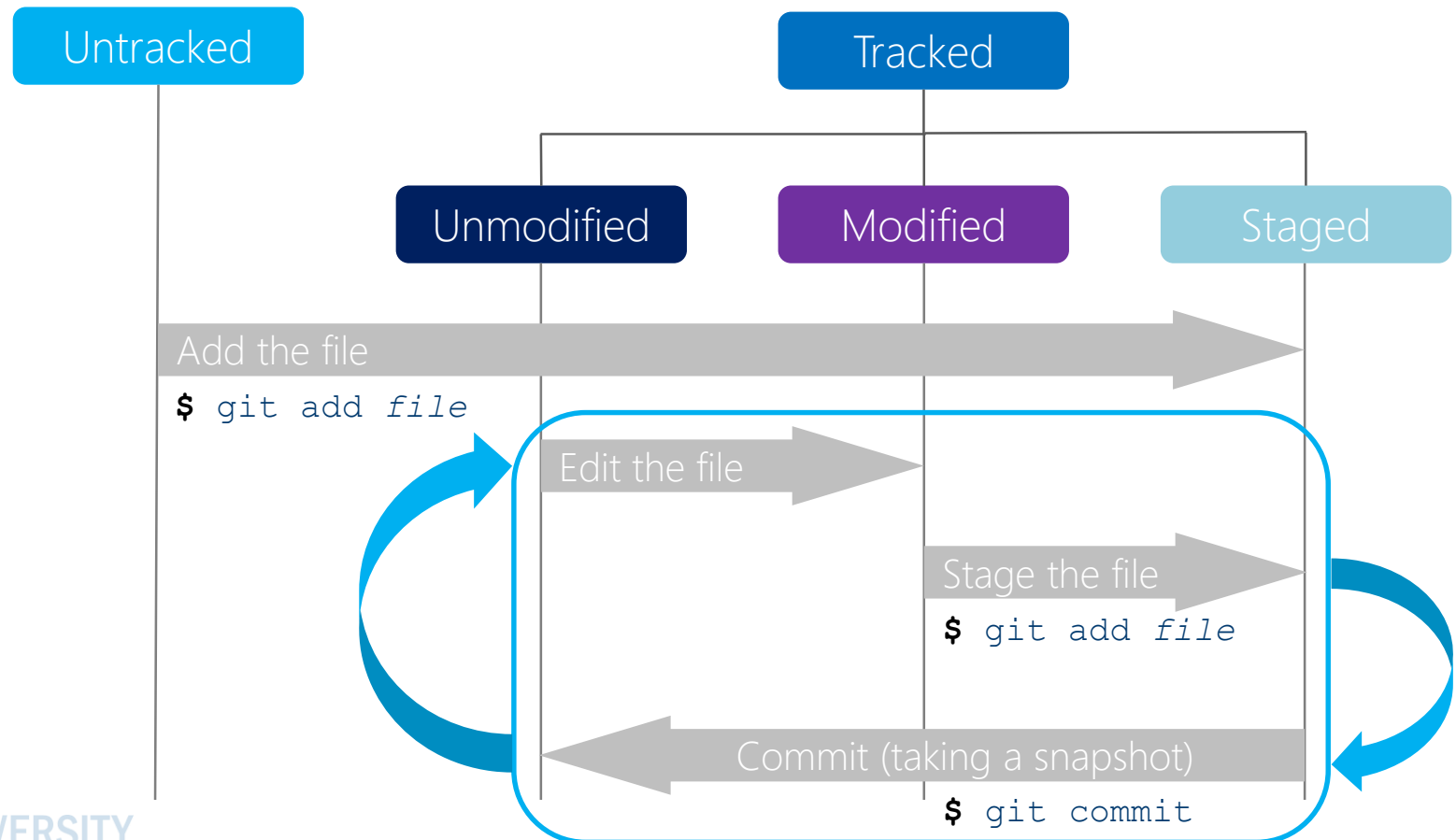
The Life Cycle of the File Status



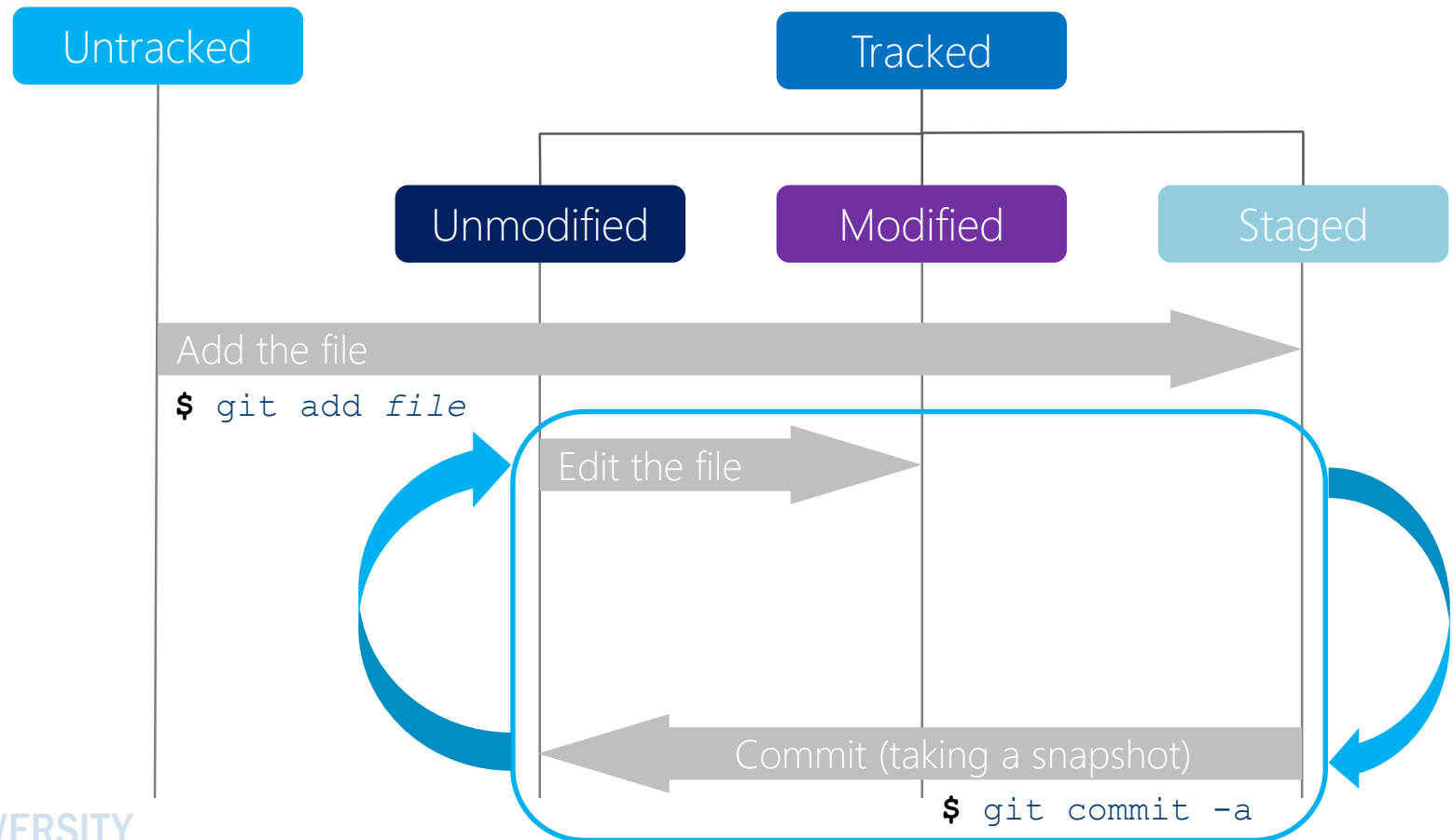
The Life Cycle of the File Status



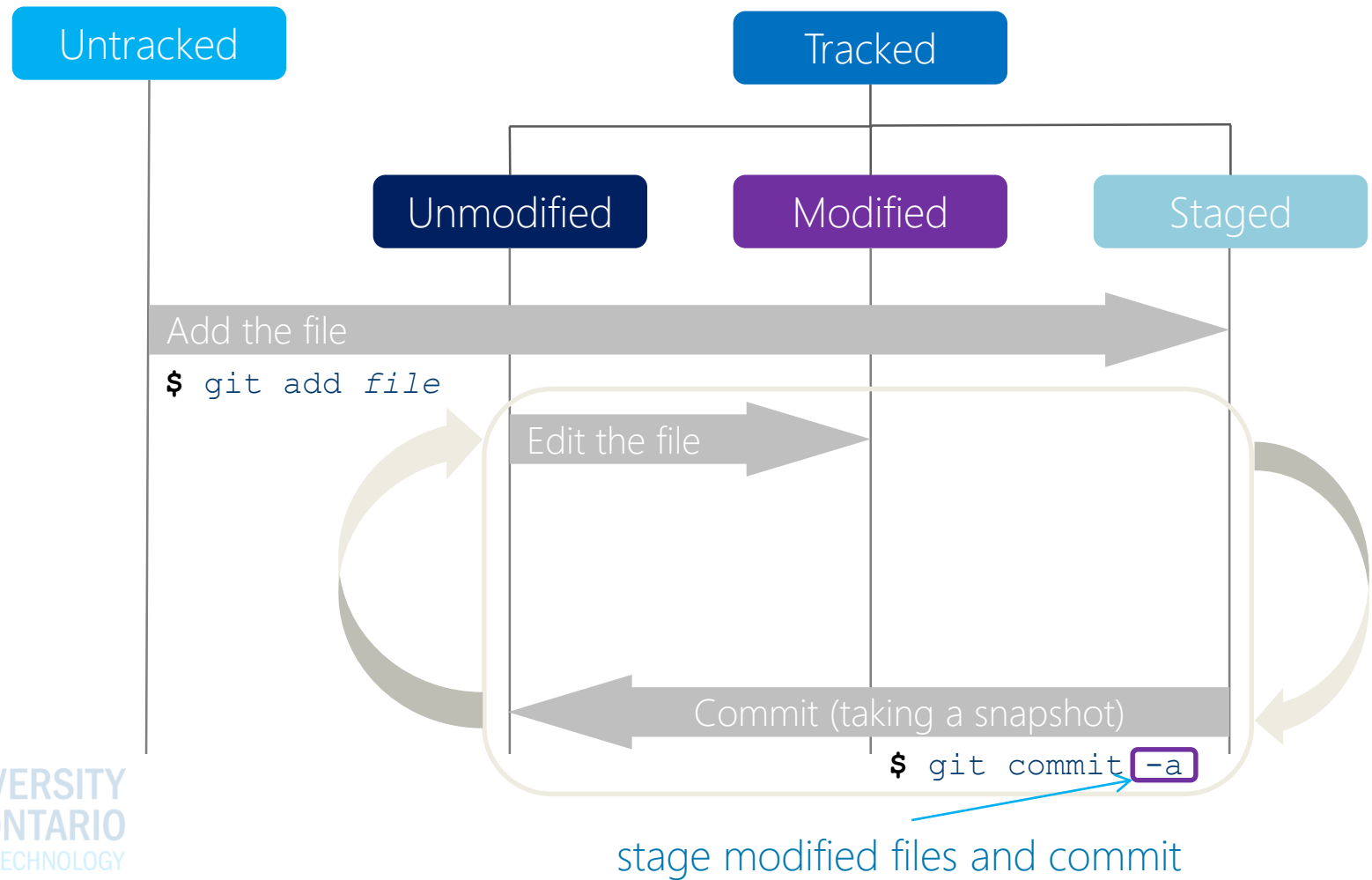
The Life Cycle of the File Status



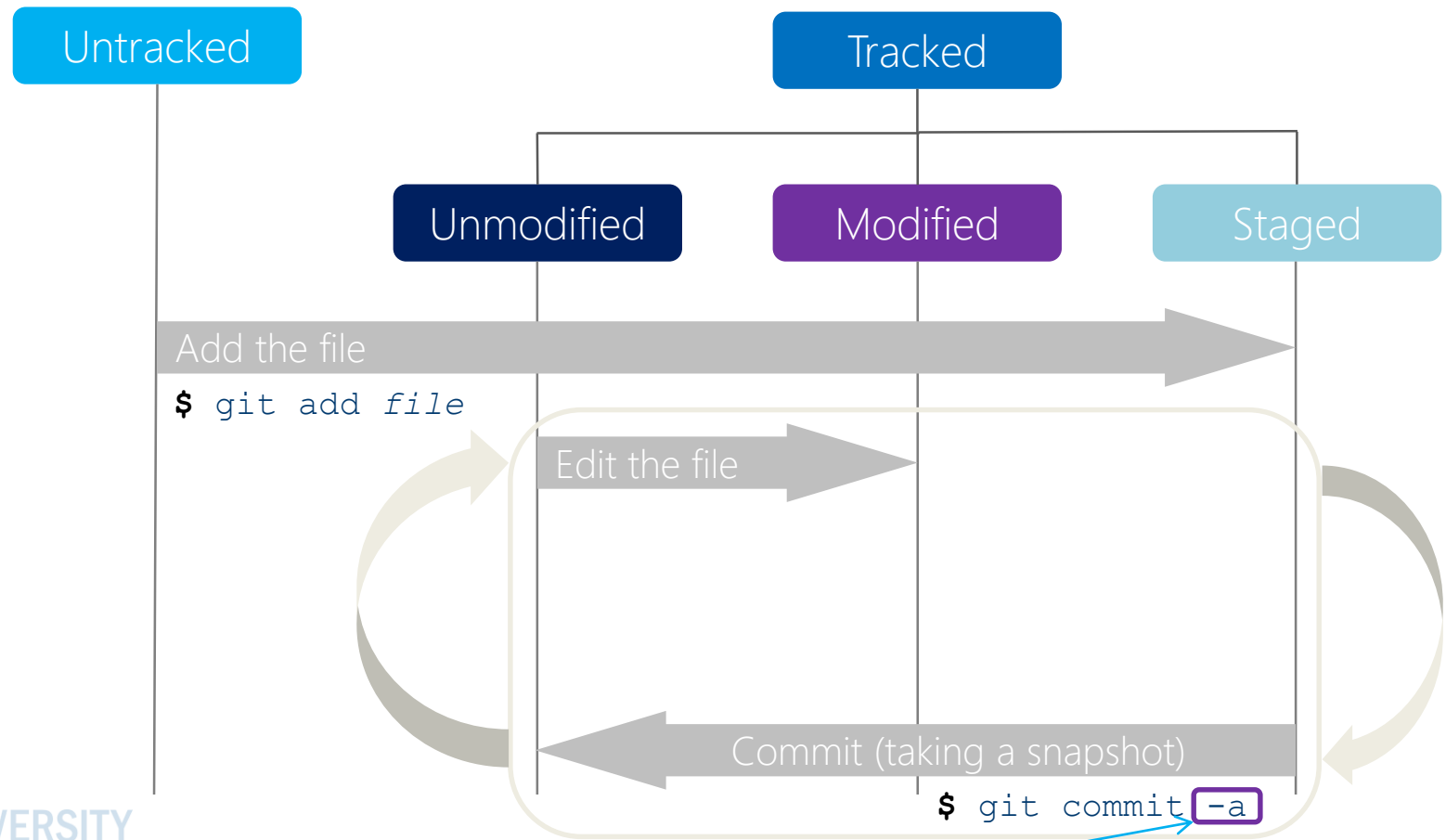
The Life Cycle of the File Status



The Life Cycle of the File Status

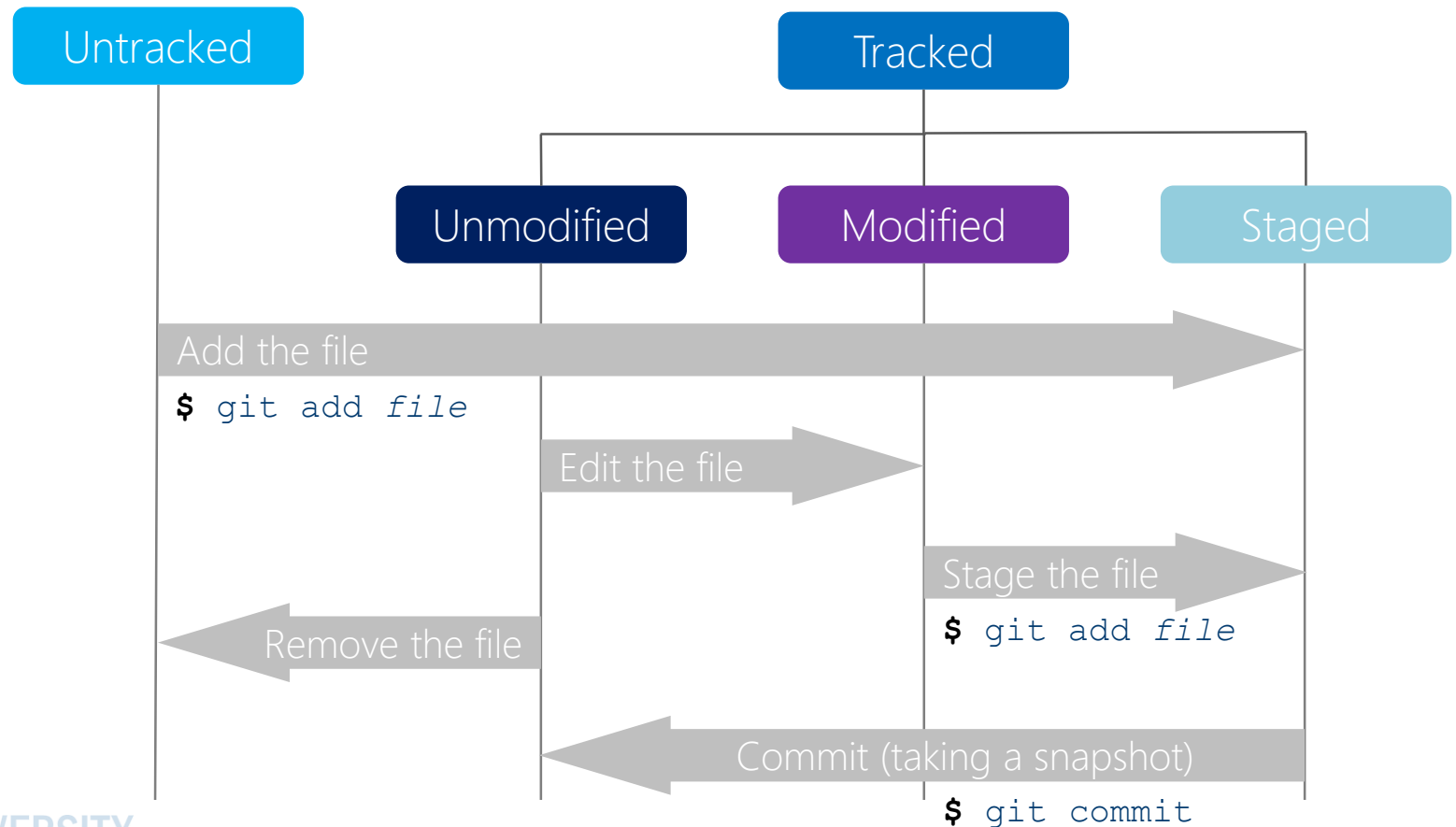


The Life Cycle of the File Status

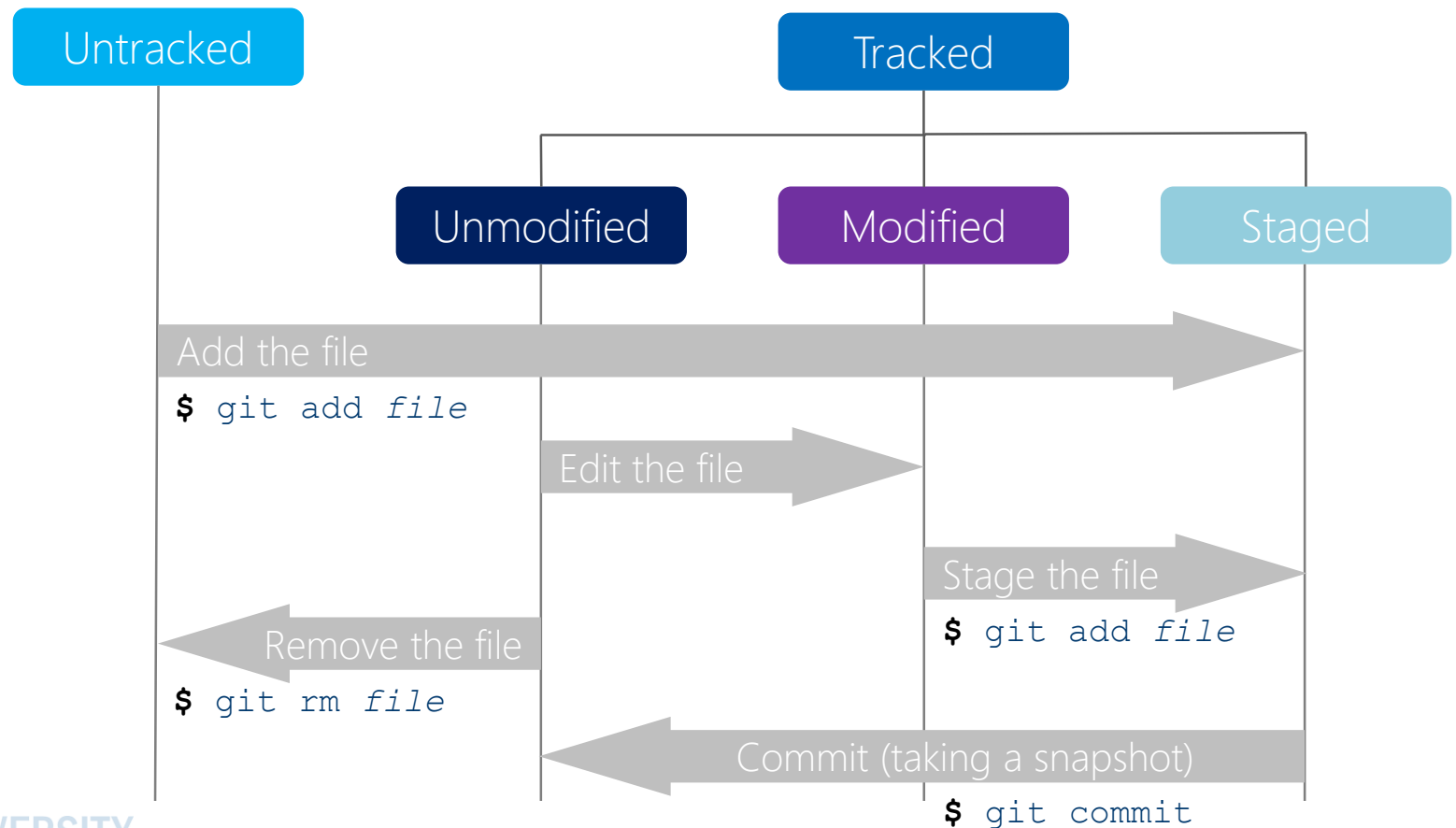


in other words, skips staging area

The Life Cycle of the File Status



The Life Cycle of the File Status



Getting Help

\$ _

Getting Help

```
$ git
```

```
$ _
```

Getting Help

```
$ git
```

```
$ git help <verb>
```

```
$ _
```

Getting Help

```
$ git
```

```
$ git help <verb>
```

```
$ _
```



replace this with *config* and
see the output

Getting Help

```
$ git
```

```
$ git help <verb>
```

```
$ git <verb> --help
```

```
$ _
```

Getting Help

```
$ git
```

```
$ git help <verb>
```

```
$ git <verb> --help
```

```
$ man git-<verb>
```

```
$ _
```

How to Unstage File(s)

\$ _

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ _
```

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ _
```

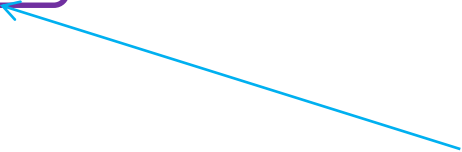


the HEAD is a pointer that holds your position within all your different commits

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ _
```



the HEAD is a pointer that holds your position within all your different commits
by default HEAD points to your most recent commit

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ _
```



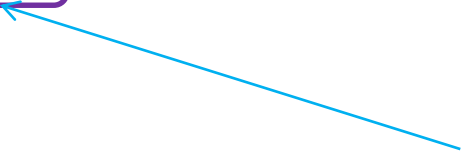
the HEAD is a pointer that holds your position within all your different commits
by default HEAD points to your most recent commit

it can be used as a quick way to reference that commit without having to look up the SHA

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ _
```



the HEAD is a pointer that holds your position within all your different commits
by default HEAD points to your most recent commit

it can be used as a quick way to reference that commit without having to look up the SHA
SHA (Secure Hash Algorithm), uniquely specifies a commit

How to Unstage File(s)

```
$ git reset HEAD <file>...
```

```
$ git rm --cached <file>...
```

```
$ _
```

Undo

\$ _

Undo

```
$ git checkout -- <file>...
```

```
$ _
```

Removing Files

\$ _

Removing Files

```
$ git rm <file>...
```

```
$ _
```


Removing Files

```
$ git rm <file>...
```

```
$ git rm '*.log'
```

```
$ _
```

Removing Files

```
$ git rm <file>...
```

```
$ git rm '*'.log'
```

```
$ _
```



prevents shell from expanding
wildcards

Removing Files

```
$ git rm <file>...
```

```
$ git rm '*.log'
```

```
$ git rm \*.log
```

```
$ _
```

Removing Files

```
$ git rm <file>...  
$ git rm '*.log'  
$ git rm \*.log  
$ _
```

another way to achieve the same
result

Moving Files

\$ _

Moving Files

```
$ git mv <file_from> <file_to>
```

```
$ _
```

Online Resource



Continue the *Git in 15 min*

Blackboard > Course Content > Week 4 (Oct. 5-9)
> Monday Oct. 5 > Git in 15 min

Online Resource



Continue the *Git in 15 min*

Blackboard > Course Content > Week 4 (Oct. 5-9)
> Monday Oct. 5 > Git in 15 min

From step 1.10 to the end (about 9 min)

Checkpoint 8



- Please complete the *What is Error Survey Results*:
 - Blackboard > Course Content > Week 4 (Oct. 5-9) > Monday Oct. 5 > Checkpoint 8



Clone a Repository from Elsewhere

\$ _

Clone a Repository from Elsewhere

```
$ git clone URL
```

```
$ _
```

Clone a Repository from Elsewhere

```
$ git clone URL
```

```
$ git clone URL mypath
```

```
$ _
```

Typical Workflow

\$ _

Typical Workflow

```
$ git status
```

```
$ _
```

Typical Workflow

```
$ git status
```

```
$ git pull
```

```
$ _
```

Typical Workflow

```
$ git status
```

```
$ git pull
```

```
$ git status
```

```
$ _
```

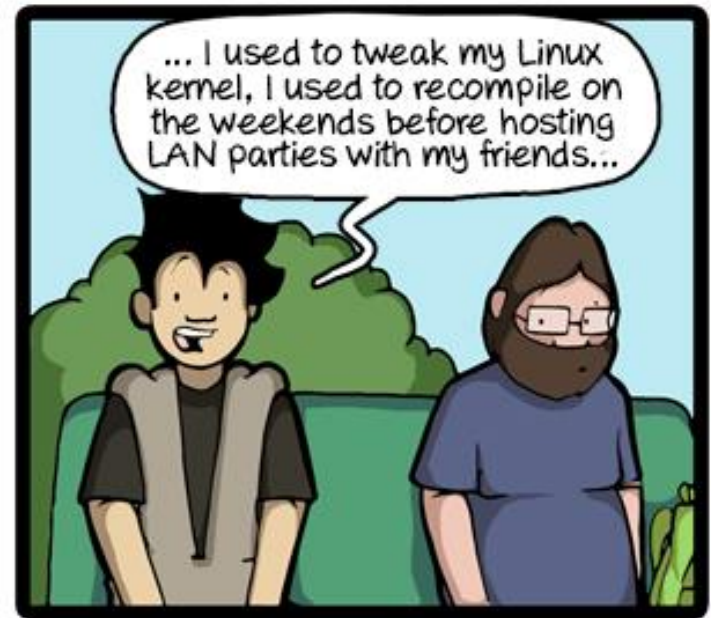

Typical Workflow

```
$ git status  
$ git pull  
$ git status  
$ git commit -a -m "what I did"  
$ _
```

Typical Workflow

```
$ git status  
$ git pull  
$ git status  
$ git commit -a -m "what I did"  
$ git push  
$ _
```

The Most Difficult 'Hello World' of Your Life!



Checkpoint 9



- Follow the instructions in:
 - Blackboard > Course Content > Week 4 (Oct. 5-9) > Monday Oct. 5 > Checkpoint 9



Checkpoint 10



- Follow the instructions in:
 - Blackboard > Course Content > Week 4 (Oct. 5-9) > Monday Oct. 5 > Checkpoint 10

