

What is version control?

- Track changes within text-based files
- Who, when, what

What is version control?

- Track changes within text-based files
- Who, when, what

Why version control?

- Track changes within text-based files
- Who, when, what
- Indirect benefit = better organization strategies

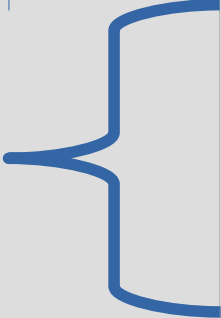
Version Control


























Version Control

It's funny 'cause it's true...

What's different between these versions?




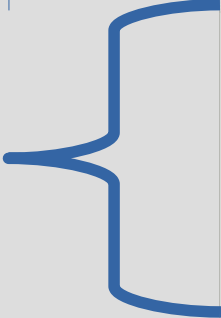
| Name | Size | Type | Modified |
|--|----------|----------|-------------|
|  SJW-Thesis-Literature Cited-DR edits.doc | 50.7 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v02.doc | 55.3 kB | Document | Jun 5 2008 |
|  SJW-Thesis-Literature Cited v03.1.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.2.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.doc | 53.2 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v04.doc | 60.9 kB | Document | Jun 21 2008 |
|  SJW-Thesis v05-DR edits v03.1.doc | 126.0 kB | Document | Mar 31 2008 |
|  SJW-Thesis v05-DR edits v03.doc | 221.2 kB | Document | Mar 25 2008 |
|  SJW-Thesis v06.1.doc | 142.3 kB | Document | Jun 7 2008 |
|  SJW-Thesis v06.doc | 133.6 kB | Document | May 31 2008 |
|  SJW-Thesis v07.1.doc | 237.1 kB | Document | Jun 9 2008 |
|  SJW-Thesis v08.doc | 242.7 kB | Document | Jun 19 2008 |
|  SJW-Thesis v09.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v09-DR edits v01.doc | 3.4 MB | Document | Jul 17 2008 |
|  SJW-Thesis v09-DR edits v02.1.doc | 5.2 MB | Document | Jul 19 2008 |
|  SJW-Thesis v09-DR edits v02.doc | 3.4 MB | Document | Jul 19 2008 |
|  SJW-Thesis v10.1.doc | 3.3 MB | Document | Jul 27 2008 |
|  SJW-Thesis v10.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v11.doc | 3.5 MB | Document | Jul 27 2008 |
|  SJW-Thesis v12.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v13.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v14.doc | 3.5 MB | Document | Jul 30 2008 |
|  SJW-Thesis v15.doc | 4.4 MB | Document | Jul 30 2008 |
| SJW-Thesis v16.doc | 4.4 MB | Document | Jul 30 2008 |
























Version Control

It's funny 'cause it's true...

What's different between these versions?

Who actually made the most recent edits?



| Name | Size | Type | Modified |
|--|----------|----------|-------------|
|  SJW-Thesis-Literature Cited-DR edits.doc | 50.7 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v02.doc | 55.3 kB | Document | Jun 5 2008 |
|  SJW-Thesis-Literature Cited v03.1.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.2.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.doc | 53.2 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v04.doc | 60.9 kB | Document | Jun 21 2008 |
|  SJW-Thesis v05-DR edits v03.1.doc | 126.0 kB | Document | Mar 31 2008 |
|  SJW-Thesis v05-DR edits v03.doc | 221.2 kB | Document | Mar 25 2008 |
|  SJW-Thesis v06.1.doc | 142.3 kB | Document | Jun 7 2008 |
|  SJW-Thesis v06.doc | 133.6 kB | Document | May 31 2008 |
|  SJW-Thesis v07.1.doc | 237.1 kB | Document | Jun 9 2008 |
|  SJW-Thesis v08.doc | 242.7 kB | Document | Jun 19 2008 |
|  SJW-Thesis v09.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v09-DR edits v01.doc | 3.4 MB | Document | Jul 17 2008 |
|  SJW-Thesis v09-DR edits v02.1.doc | 5.2 MB | Document | Jul 19 2008 |
|  SJW-Thesis v09-DR edits v02.doc | 3.4 MB | Document | Jul 19 2008 |
|  SJW-Thesis v10.1.doc | 3.3 MB | Document | Jul 27 2008 |
|  SJW-Thesis v10.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v11.doc | 3.5 MB | Document | Jul 27 2008 |
|  SJW-Thesis v12.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v13.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v14.doc | 3.5 MB | Document | Jul 30 2008 |
|  SJW-Thesis v15.doc | 4.4 MB | Document | Jul 30 2008 |
| SJW-Thesis v16.doc | 4.4 MB | Document | Jul 30 2008 |

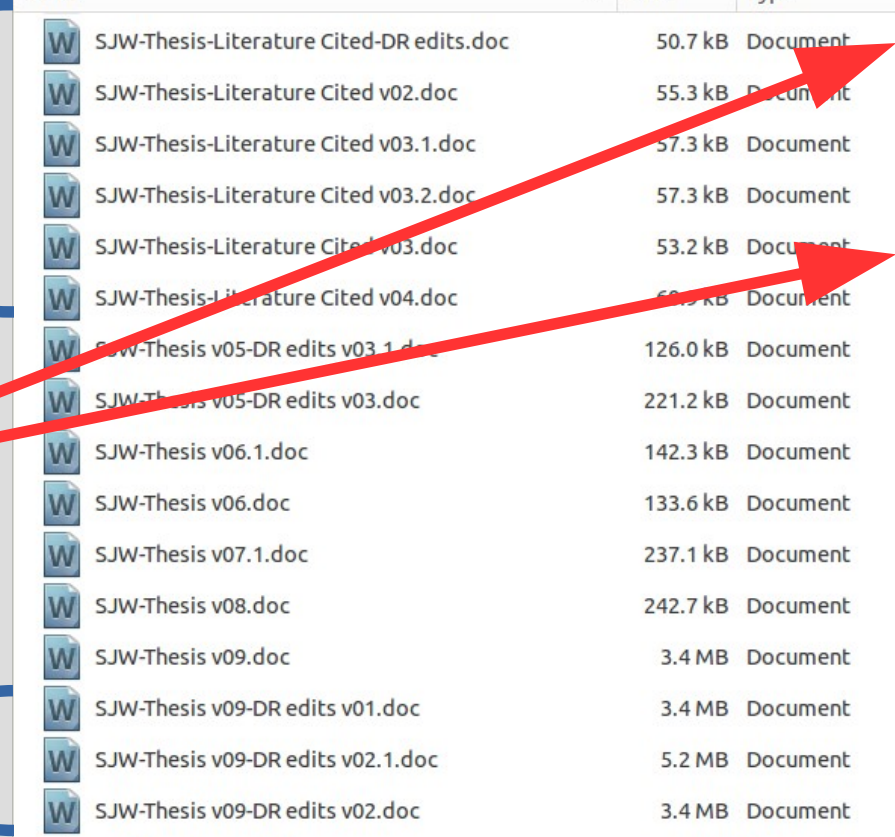
Version Control
























It's funny 'cause it's true...

What's different between these versions?

Which one of these is the most recent??!










Who actually made the edits?



| Name | Size | Type | Modified |
|--|----------|----------|-------------|
|  SJW-Thesis-Literature Cited-DR edits.doc | 50.7 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v02.doc | 55.3 kB | Document | Jun 5 2008 |
|  SJW-Thesis-Literature Cited v03.1.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.2.doc | 57.3 kB | Document | Jun 8 2008 |
|  SJW-Thesis-Literature Cited v03.doc | 53.2 kB | Document | May 31 2008 |
|  SJW-Thesis-Literature Cited v04.doc | 58.9 kB | Document | Jun 21 2008 |
|  SJW-Thesis v05-DR edits v03.1.doc | 126.0 kB | Document | Mar 31 2008 |
|  SJW-Thesis v05-DR edits v03.doc | 221.2 kB | Document | Mar 25 2008 |
|  SJW-Thesis v06.1.doc | 142.3 kB | Document | Jun 7 2008 |
|  SJW-Thesis v06.doc | 133.6 kB | Document | May 31 2008 |
|  SJW-Thesis v07.1.doc | 237.1 kB | Document | Jun 9 2008 |
|  SJW-Thesis v08.doc | 242.7 kB | Document | Jun 19 2008 |
|  SJW-Thesis v09.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v09-DR edits v01.doc | 3.4 MB | Document | Jul 17 2008 |
|  SJW-Thesis v09-DR edits v02.1.doc | 5.2 MB | Document | Jul 19 2008 |
|  SJW-Thesis v09-DR edits v02.doc | 3.4 MB | Document | Jul 19 2008 |
|  SJW-Thesis v10.1.doc | 3.3 MB | Document | Jul 27 2008 |
|  SJW-Thesis v10.doc | 3.4 MB | Document | Jun 22 2008 |
|  SJW-Thesis v11.doc | 3.5 MB | Document | Jul 27 2008 |
|  SJW-Thesis v12.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v13.doc | 4.5 MB | Document | Jul 29 2008 |
|  SJW-Thesis v14.doc | 3.5 MB | Document | Jul 30 2008 |
|  SJW-Thesis v15.doc | 4.4 MB | Document | Jul 30 2008 |
| SJW-Thesis v16.doc | 4.4 MB | Document | Jul 30 2008 |










It's funny 'cause it's true...

Wish I had thought of this earlier...

| Name | |
|---|--|
|  | ACKNOWLEDGEMENTS Fix SJW.pdf |
|  | ABSTRACT Fix SJW.pdf |
|  | FormatManual2007_web.pdf |
|  | LITERATURE CITED Fix SJWpdf.pdf |
|  | INTRODUCTION Fix SJW.pdf |
|  | MATERIALS AND METHODS Fix p36-38 SJW.pdf |
|  | White, S - 7-2-2008.pdf |
|  | TABLE OF CONTENTS Fix SJW.pdf |
|  | TITLE PAGE Fix SJW.pdf |

It's funny 'cause it's true...

Wish I had thought of this earlier...

| Name | |
|---|--|
|  | ACKNOWLEDGEMENTS Fix SJW.pdf |
|  | ABSTRACT Fix SJW.pdf |
|  | FormatManual2007_web.pdf |
|  | LITERATURE CITED Fix SJWpdf.pdf |
|  | INTRODUCTION Fix SJW.pdf |
|  | MATERIALS AND METHODS Fix p36-38 SJW.pdf |
|  | White, S - 7-2-2008.pdf |
|  | TABLE OF CONTENTS Fix SJW.pdf |
|  | TITLE PAGE Fix SJW.pdf |

Project segmentation has significant benefits for version control, too!

Let's *git* in to it... (cue collective groan)

- Verify git is installed

```
$which git
```

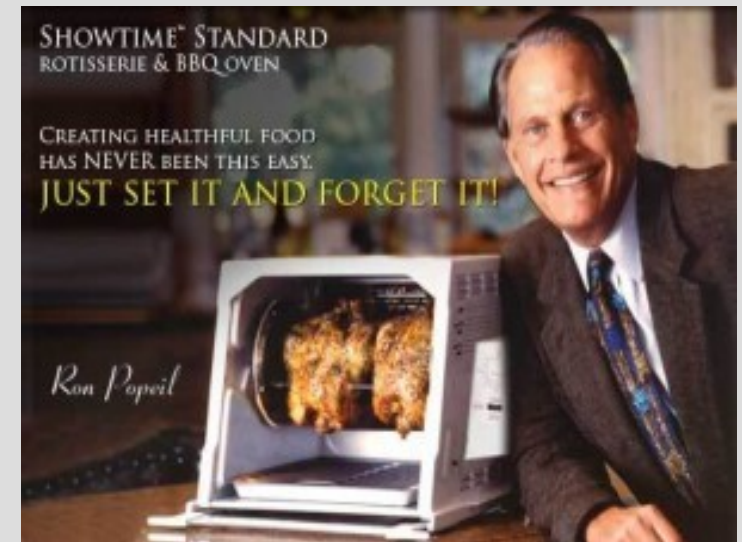
Let's *git* in to it... (cue collective groan)

- Verify git is installed

```
$which git  
/usr/bin/git
```

Let's *git* in to it... (cue collective groan)

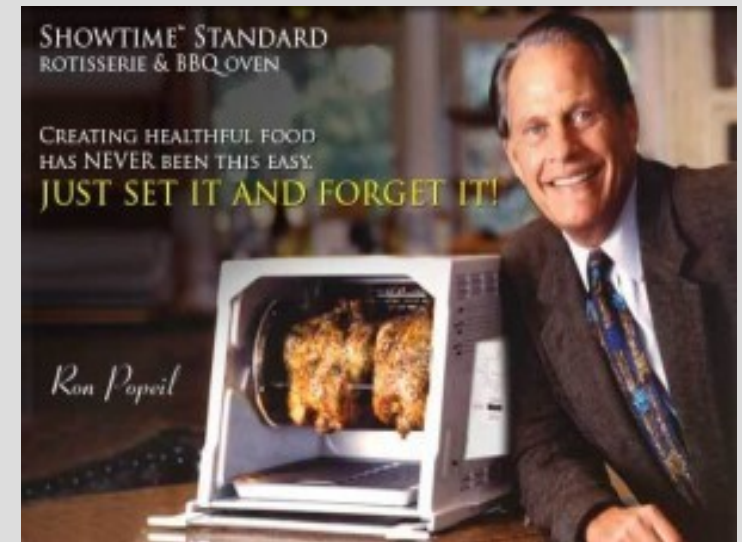
- Configure git



```
$git config --global user.name "<your name>"
```

Let's *git* in to it... (cue collective groan)

- Configure git



```
$git config --global user.name "<your name>"  
$git config --global user.email "<your email>"
```

Let's *git* in to it... (cue collective groan)

- Configure git



```
$git config --global user.name "<your name>"  
$git config --global user.email "<your email>"  
$git config --global color.ui "auto"
```

Let's *git* in to it... (cue collective groan)

- Configure git



```
$git config --global user.name "<your name>"
$git config --global user.email "<your email>"
$git config --global color.ui "auto"
$git config --global core.editor "nano"
```


Let's *really git* in to it... (yep, still using this pun)

- Change to home directory
- Make a new directory called: manuscript
- Change to manuscript directory

Let's *really git* in to it... (yep, still using this pun)

- Initialize our repository

```
$git init
```

Let's *really git* in to it... (yep, still using this pun)

- Initialize our repository

```
$git init
```



Repository (repo): Fancy name for folder monitored by git

Let's *really git* in to it... (yep, still using this pun)

- Initialize our repository

```
$git init
```

What happened? Let's see...

```
$ls
```

Let's *really git* in to it... (yep, still using this pun)

- Initialize our repository

```
$git init
```

What happened? Let's see...

```
$ls  
$ls -a  
.  ..  .git
```

Let's *really git* in to it... (yep, still using this pun)

- Initialize our repository

```
$git init
```

What happened? Let's see...

```
$ls  
$ls -a  
. .. .git
```

Hidden folder! Stores all file versions, keeps repo neat

Let's *really git* in to it... (yep, still using this pun)

- Verify git is set up

```
$git status
```

Let's *really git* in to it... (yep, still using this pun)

- Verify git is set up

```
$git status  
On branch master
```

```
Initial commit
```

```
nothing to commit (create/copy files and  
use "git add" to track)
```

Let's *really git* in to it... (yep, still using this pun)

- Verify git is set up

```
$git status
On branch master

Initial commit

nothing to commit (create/copy files and
use "git add" to track)
```

Great tip! Let's do that.

Let's *really git* in to it... (yep, still using this pun)

- Make a file for git to track

```
$nano paper.txt
```

- Enter some text in the file, save
- Verify our file was created

```
$ls  
paper.txt
```

- Verify content of our file

```
$cat paper.txt  
INTRODUCTION
```

Let's *really git* in to it... (yep, still using this pun)

- Check repo ***status***

```
$git status
On branch master

Initial commit

Untracked files:
  (use "git add <file>..." to include in
what will be committed)

  paper.txt

nothing added to commit but untracked files
present (use "git add" to track)
```

Let's *really git* in to it... (yep, still using this pun)

- Start tracking our file

```
$git add paper.txt
```


Let's *really git* in to it... (yep, still using this pun)

- Start tracking our file

```
$git add paper.txt
```

- How do we check the ***status*** of our repository?

Let's *really git* in to it... (yep, still using this pun)

- Start tracking our file

```
$git add paper.txt
```

- How do we check the ***status*** of our repository?

```
$git status
```

Let's *really git* in to it... (yep, still using this pun)

```
$git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   paper.txt
```

Let's *really git* in to it... (yep, still using this pun)

```
$git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   paper.txt
```

Excellent! Git detects that our file has changed.

Let's *really git* in to it... (yep, still using this pun)

- ***Commit*** our changes

```
$git commit -m "add introduction section title"
```

Let's *really git* in to it... (yep, still using this pun)

- **Commit** our changes to memory

```
$git commit -m "add introduction section title"  
[master (root-commit) f2ac8a2]  
add introduction section title  
 1 file changed, 1 insertion(+)  
create mode 100644 paper.txt
```

Great! But, what does this all mean?

Let's *really git* in to it... (yep, still using this pun)

```
$git commit -m "add introduction section title"  
[master (root-commit) f2ac8a2]  
add introduction section title  
1 file changed, 1 insertion(+)  
create mode 100644 paper.txt
```

-m: Message flag for **commit** command

Let's *really git* in to it... (yep, still using this pun)

```
$git commit -m "add introduction section title"  
[master (root-commit) f2ac8a2]  
add introduction section title  
1 file changed, 1 insertion(+)  
create mode 100644 paper.txt
```

-m: Message flag for **commit** command

f2ac8a2: Short unique revision ID

Let's *really git* in to it... (yep, still using this pun)

```
$git commit -m "add introduction section title"  
[master (root-commit) f2ac8a2]  
add introduction section title  
1 file changed, 1 insertion(+)  
create mode 100644 paper.txt
```

-m: Message flag for **commit** command

f2ac8a2: Short unique revision ID

Message: The message associated with this revision

Let's *really git* in to it... (yep, still using this pun)

```
$git commit -m "add introduction section title"  
[master (root-commit) f2ac8a2]  
add introduction section title  
1 file changed, 1 insertion(+)  
create mode 100644 paper.txt
```

-m: Message flag for **commit** command

f2ac8a2: Short unique revision ID

Message: The message associated with this revision

Summary: # files changed/type of change(s)

Let's *really git* in to it... (yep, still using this pun)

What's the ***status*** of our repo now?

```
$git status
On branch master
nothing to commit, working directory clean
```

Excellent! All changes have been accounted for.

How do we see a ***log*** (history) of any changes?

Let's *really git* in to it... (yep, still using this pun)

- See the **log** of any committed changes

```
$git log
commit f2ac8a26dc0eac59aff4ee29a50165a4adc6cd5e
Author: Sam <samwhite@uw.edu>
Date:    Sun Jan 11 21:45:14 2015 -0800

    add introduction section title
```

Let's *really git* in to it... (yep, still using this pun)

```
$git log
commit f2ac8a26dc0eac59aff4ee29a50165a4adc6cd5e
Author: Sam <samwhite@uw.edu>
Date:   Sun Jan 11 21:45:14 2015 -0800

    add introduction section title
```

commit: Full unique revision ID

Let's *really git* in to it... (yep, still using this pun)

```
$git log
commit f2ac8a26dc0eac59aff4ee29a50165a4adc6cd5e
Author: Sam <samwhite@uw.edu>
Date:   Sun Jan 11 21:45:14 2015 -0800

    add introduction section title
```

commit: Full unique revision ID (only need short one, though)

Let's *really git* in to it... (yep, still using this pun)

```
$git log
commit f2ac8a26dc0eac59aff4ee29a50165a4adc6cd5e
Author: Sam <samwhite@uw.edu>
Date:    Sun Jan 11 21:45:14 2015 -0800

    add introduction section title
```

commit: Full unique revision ID (only need short one, though)

Author: Who made **commit**

Date: Date/time of **commit**

Message: **Commit** message

Let's *really git* in to it... (yep, still using this pun)

- Make changes to our file

```
$nano paper.txt
$cat paper.txt
INTRODUCTION
This is just the beginning.
```

Let's *really git* in to it... (yep, still using this pun)

- Make changes to our file

```
$nano paper.txt
$cat paper.txt
INTRODUCTION
This is just the beginning.
```

- Check ***status*** of our repo

```
$git status
```

Let's *really git* in to it... (yep, still using this pun)

- Check the ***differences*** between current and committed versions

```
$git diff
```

Let's *really git* in to it... (yep, still using this pun)

- Check the ***differences*** between current and committed version

```
$git diff
diff --git a/paper.txt b/paper.txt
index 1797975..eb2450a 100644
--- a/paper.txt
+++ b/paper.txt
@@ -1,2 @@
  INTRODUCTION
+This is just the beginning.
```

Let's *really git* in to it... (yep, still using this pun)

```
$git diff
diff --git a/paper.txt b/paper.txt
index 1797975..eb2450a 100644
--- a/paper.txt
+++ b/paper.txt
@@ -1,2 @@
  INTRODUCTION
+This is just the beginning.
```

1797975..eb2450a: Short unique revision IDs

Let's *really git* in to it... (yep, still using this pun)

```
$git diff
diff --git a/paper.txt b/paper.txt
index 1797975..eb2450a 100644
--- a/paper.txt
+++ b/paper.txt
@@ -1,2 @@
 INTRODUCTION
+This is just the beginning.
```

1797975..eb2450a: Short unique revision IDs

+: Added line in current version

INTRODUCTION has no indicator = unchanged content

Let's *really git* in to it... (yep, still using this pun)

```
$git diff
diff --git a/paper.txt b/paper.txt
index 1797975..eb2450a 100644
--- a/paper.txt
+++ b/paper.txt
@@ -1,2 @@
 INTRODUCTION
+This is just the beginning.
```


Let's *really git* in to it... (yep, still using this pun)

- ***Commit*** our most recent changes

```
$git commit
```

Let's *really git* in to it... (yep, still using this pun)

- ***Commit*** our most recent changes

```
$git commit -m "introductory sentence"
```

DOH! We forgot to ***add*** our most recent changes first!

Let's *really git* in to it... (yep, still using this pun)

- **Commit** our most recent changes

```
$git commit -m "add first sentence to intro"
```

DOH! We forgot to **add** our most recent changes first!

```
$git add paper.txt
```

Let's *really git* in to it... (yep, still using this pun)

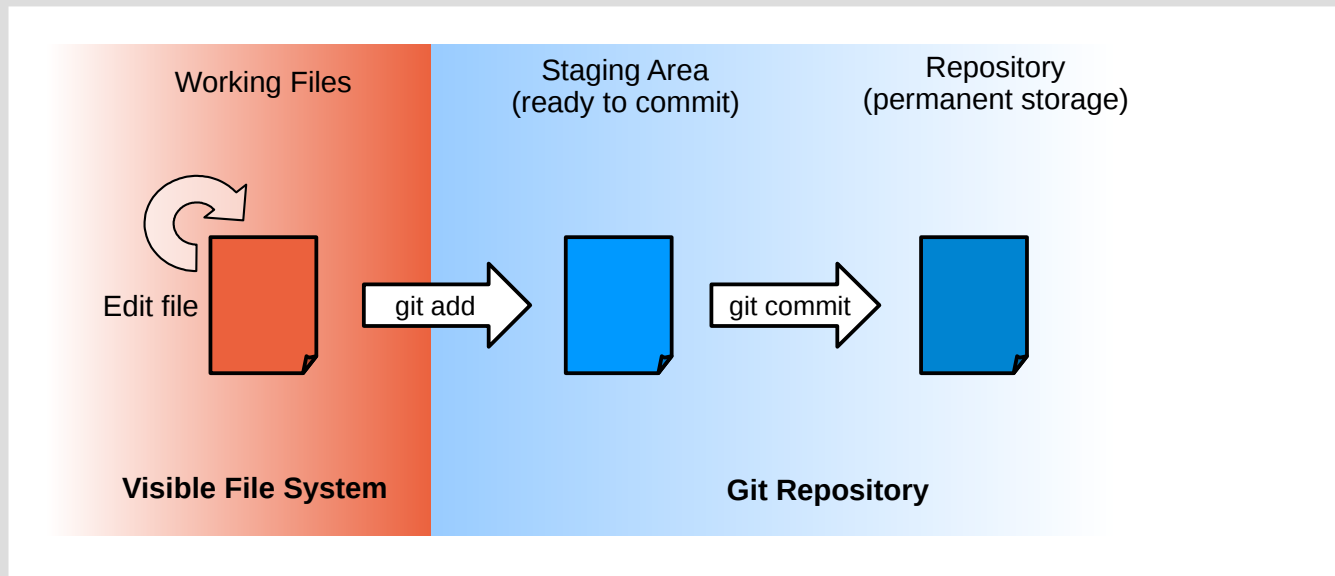
- Before committing, let's double-check the changes we made

```
$git diff  
$
```

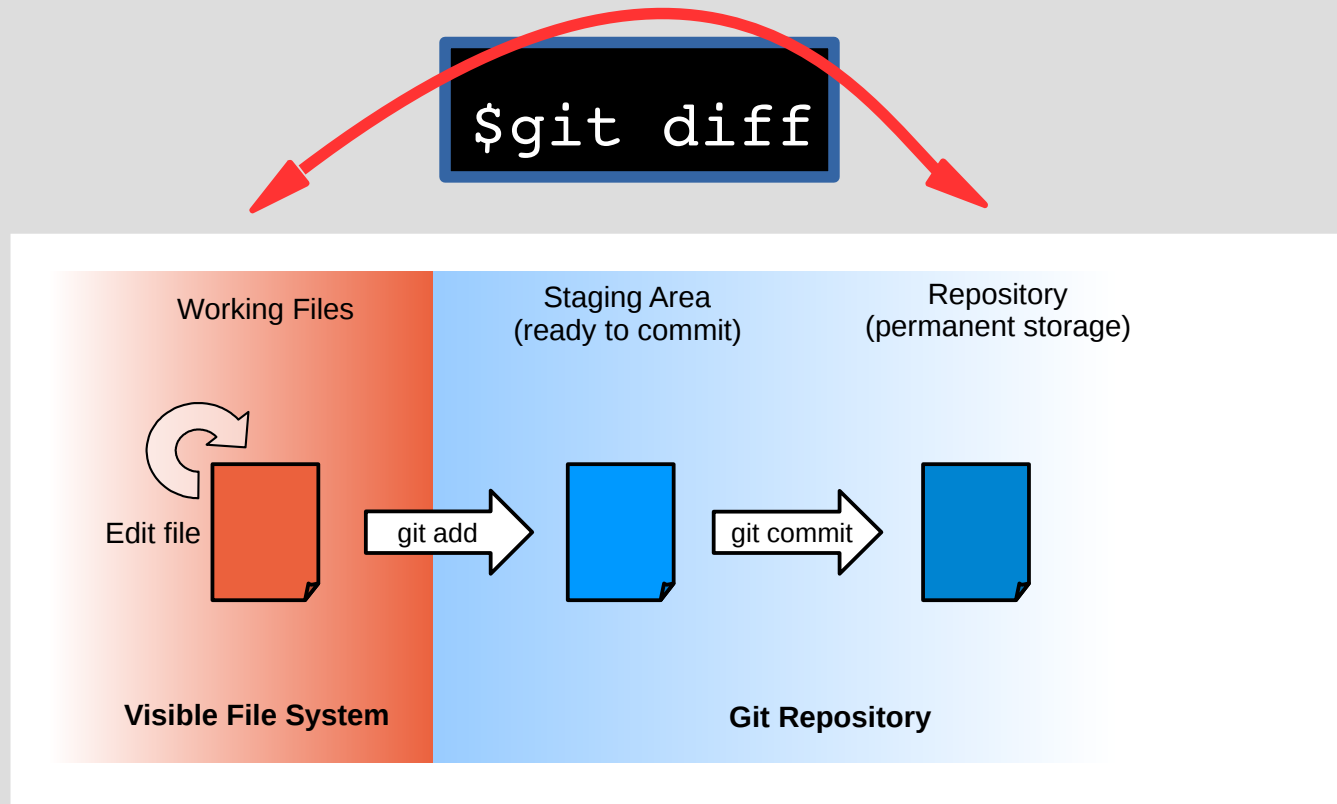
Wait, what's going on here? Where are the changes we made??

Let's *really git* in to it... (yep, still using this pun)

- The answer lies in the process...
- Remember: git manages *changes* to files



- No changes remain in our file; ***added*** to staging area

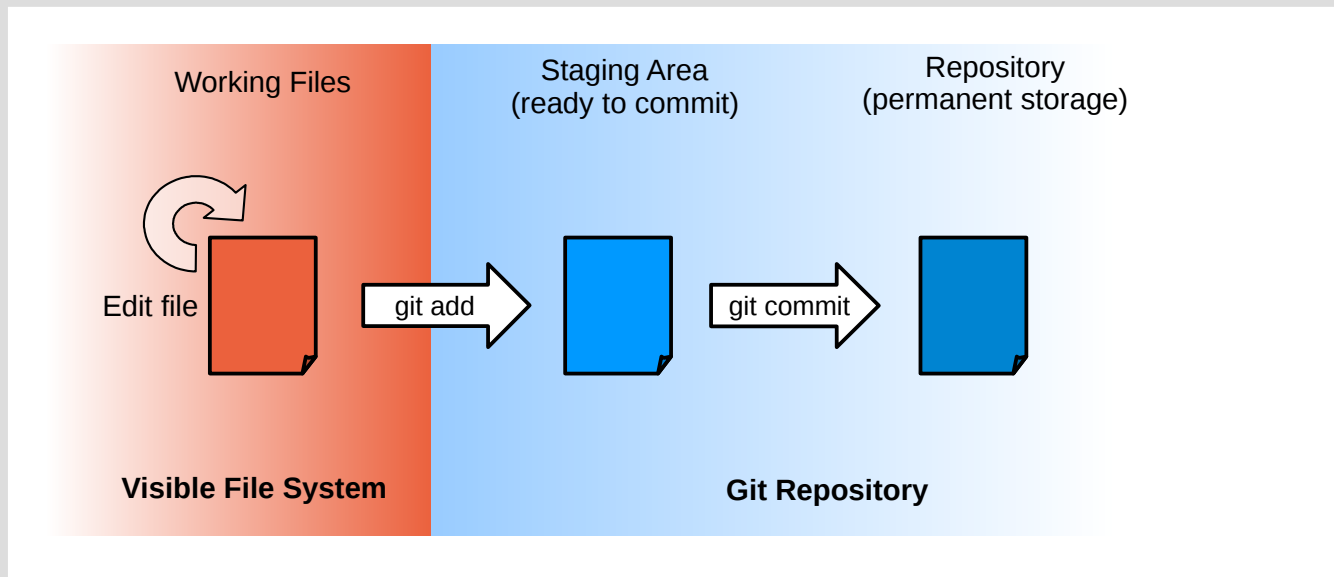




Staging area: *Changes* to files waiting to be committed



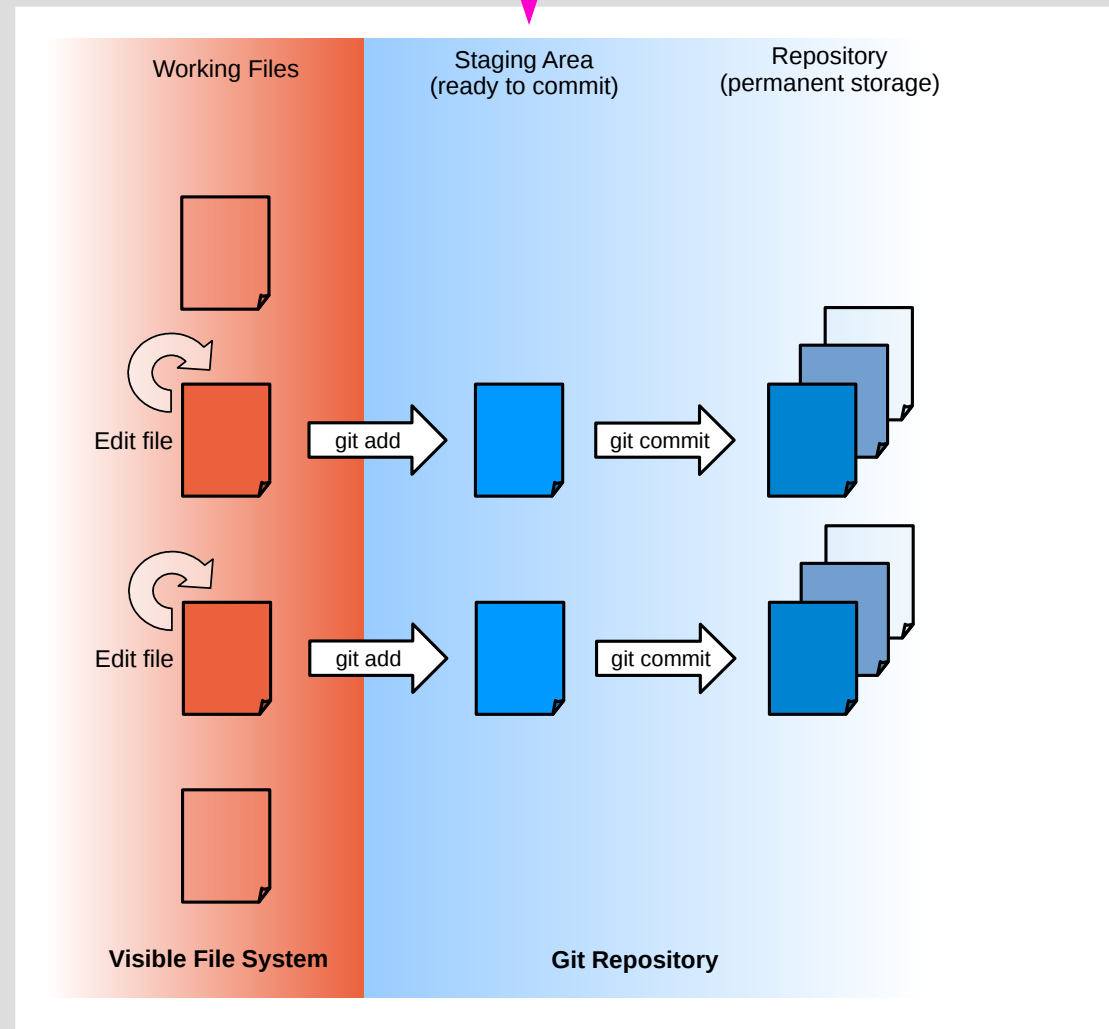
Staging area: *Changes to files waiting to be committed*



Why? What's the point? Why not just ***commit***?

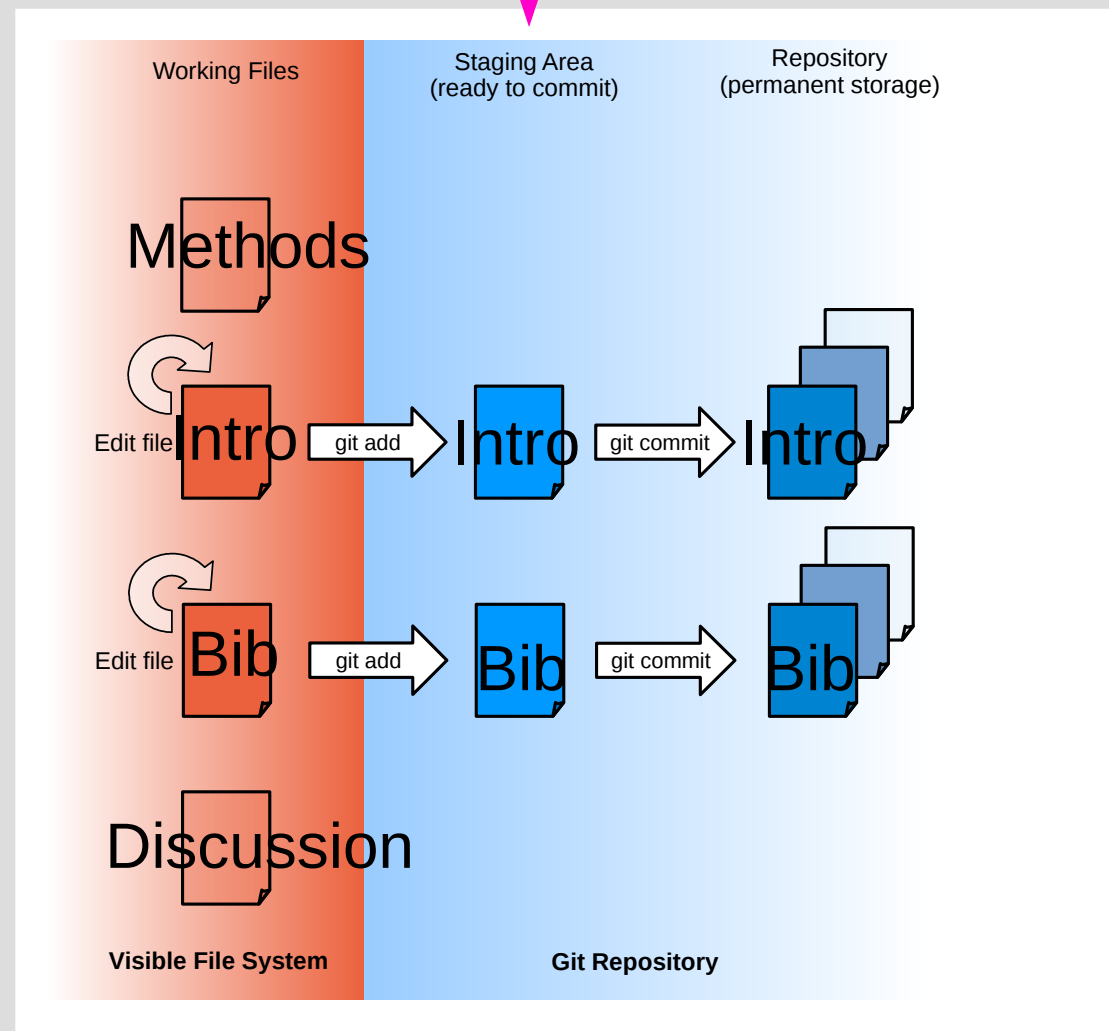
- Acts on *all changes in staging area*

```
$git commit -m "message"
```



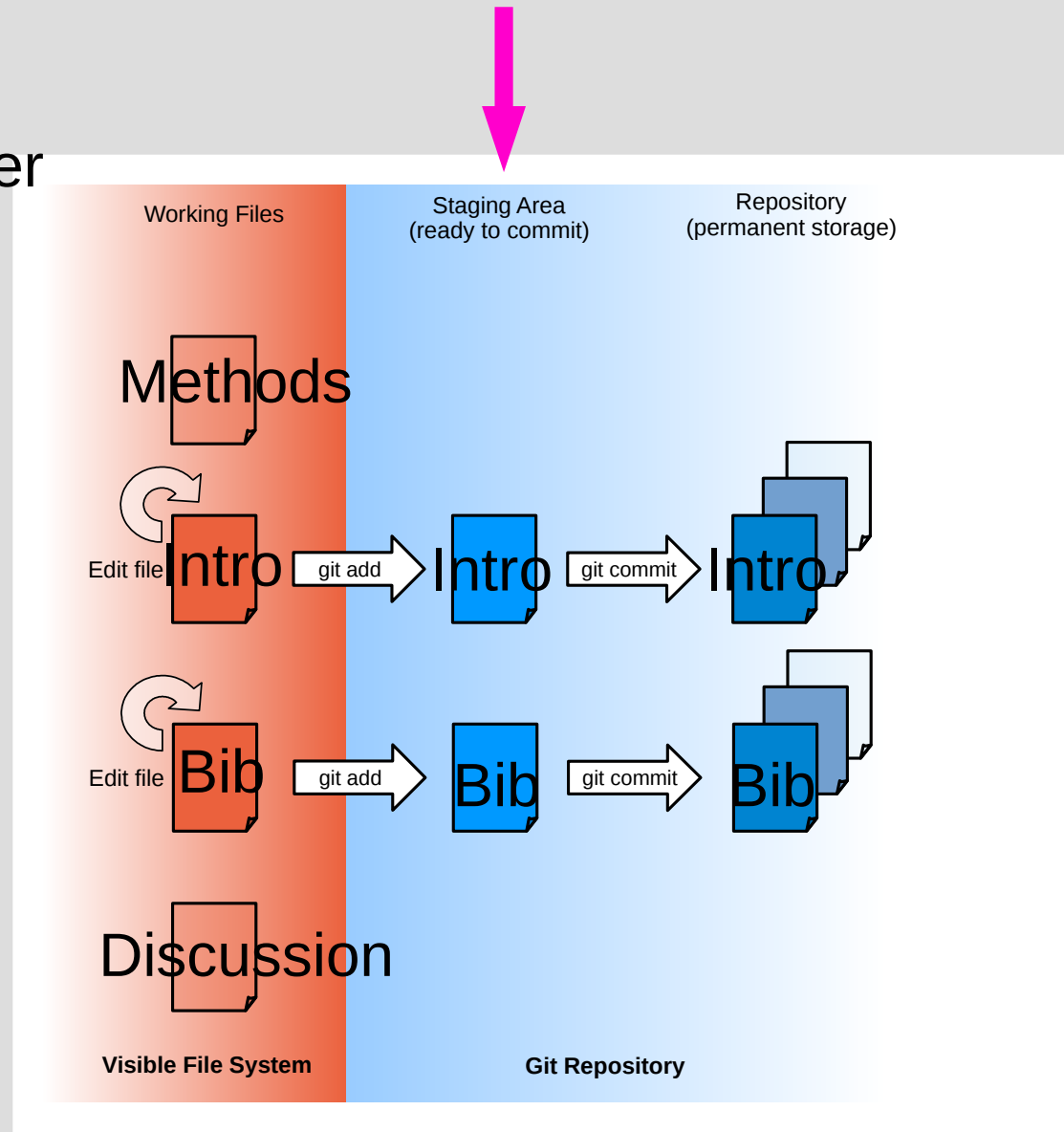
- Acts on *all changes in staging area*

```
$git commit -m "message"
```



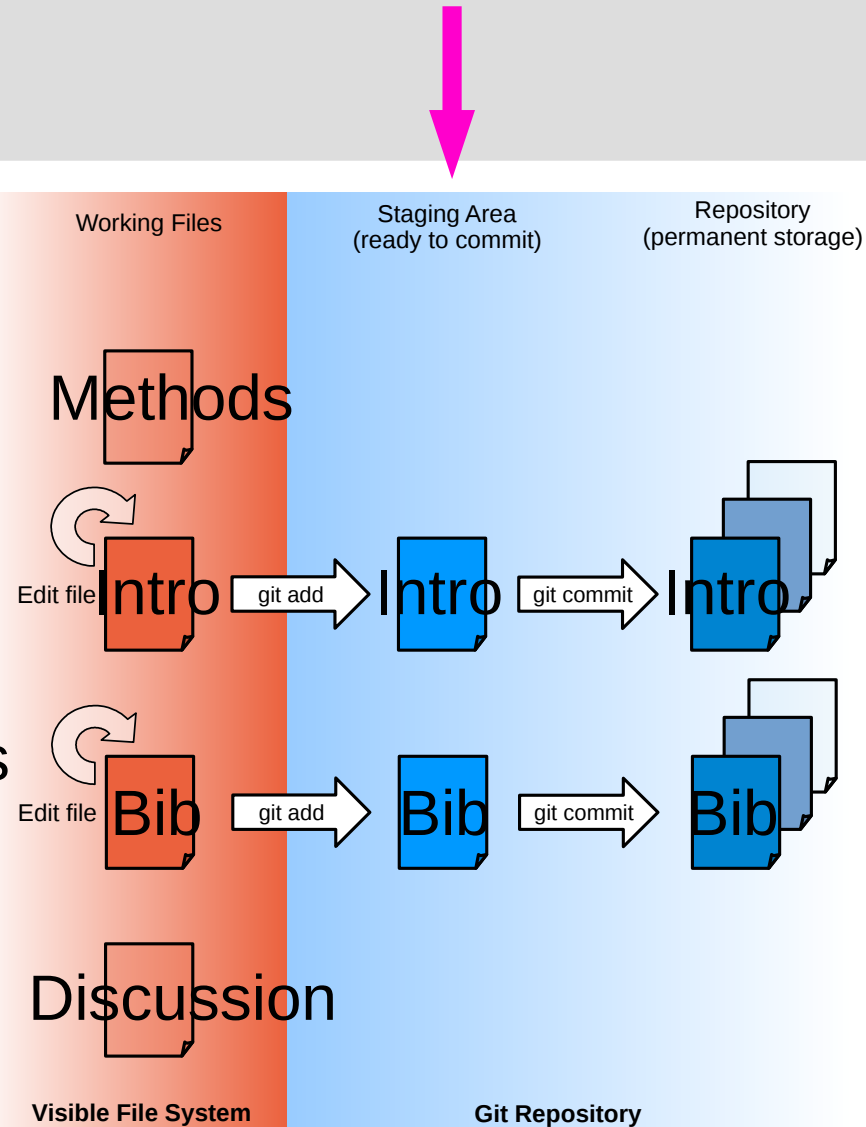
- Acts on *all changes in staging area*
 - If no buffer, all changes in all files committed together

```
$git commit -m "message"
```



- Acts on *all changes in staging area*
 - If no buffer, all changes in all files committed together
- Benefits
 - Unrelated changes separate
 - Related changes “linked”
 - Faster than individual commits

```
$git commit -m "message"
```



If I can't remember the changes I made to ***staged*** files, but *\$git diff* won't show me.

```
$git diff --staged
```

If I can't remember the changes I made to ***staged*** files, but *\$git diff* won't show me.

```
$git diff --staged
```

- Shows difference between ***staged*** file and most recent ***commit***

Go through the entire work flow

1. Add a line to our file

```
$nano paper.txt
```

2. Verify addition

```
$cat paper.txt
```


Go through the entire work flow

1. Add a line to our file

```
$nano paper.txt
```

2. Verify addition

```
$cat paper.txt
```

3. Check *diff*erences

```
$git diff
```

Go through the entire work flow

1. Add a line to our file

```
$nano paper.txt
```

2. Verify addition

```
$cat paper.txt
```

3. Check *diff*erences

```
$git diff
```

4. Stage our changes

```
$git add paper.txt
```

Go through the entire work flow

1. Add a line to our file

```
$nano paper.txt
```

2. Verify addition

```
$cat paper.txt
```

3. Check *diff*erences

```
$git diff
```

4. Stage our file

```
$git add paper.txt
```

5. Check *diff*erences

```
$git diff
```

Go through the entire work flow

6. Check *staged diff*

```
$git diff --staged
```

Go through the entire work flow

6. Check staged *diff*

```
$git diff --staged
```

7. **Commit** changes

```
$git commit -m "more intro"
```

Go through the entire work flow

6. Check staged *diff*

```
$git diff --staged
```

7. **Commit** changes

```
$git commit -m "more intro"
```

8. Check *status*

```
$git status
```

Go through the entire work flow

6. Check staged *diff*

```
$git diff --staged
```

7. **Commit** changes

```
$git commit -m "more intro"
```

8. Check *status*

```
$git status
```

9. Check change *log*

```
$git log
```

Go through the entire work flow

6. Check staged *diff*

```
$git diff --staged
```

7. **Commit** changes

```
$git commit -m "more intro"
```

8. Check *status*

```
$git status
```

9. Check change *log*

```
$git log
```


Tracking Changes: Take Home

Make Changes



Add changes



Commit changes

Handling History

```
$git log
```

- Top entry (most recent ***commit***) is the ***HEAD***

Handling History

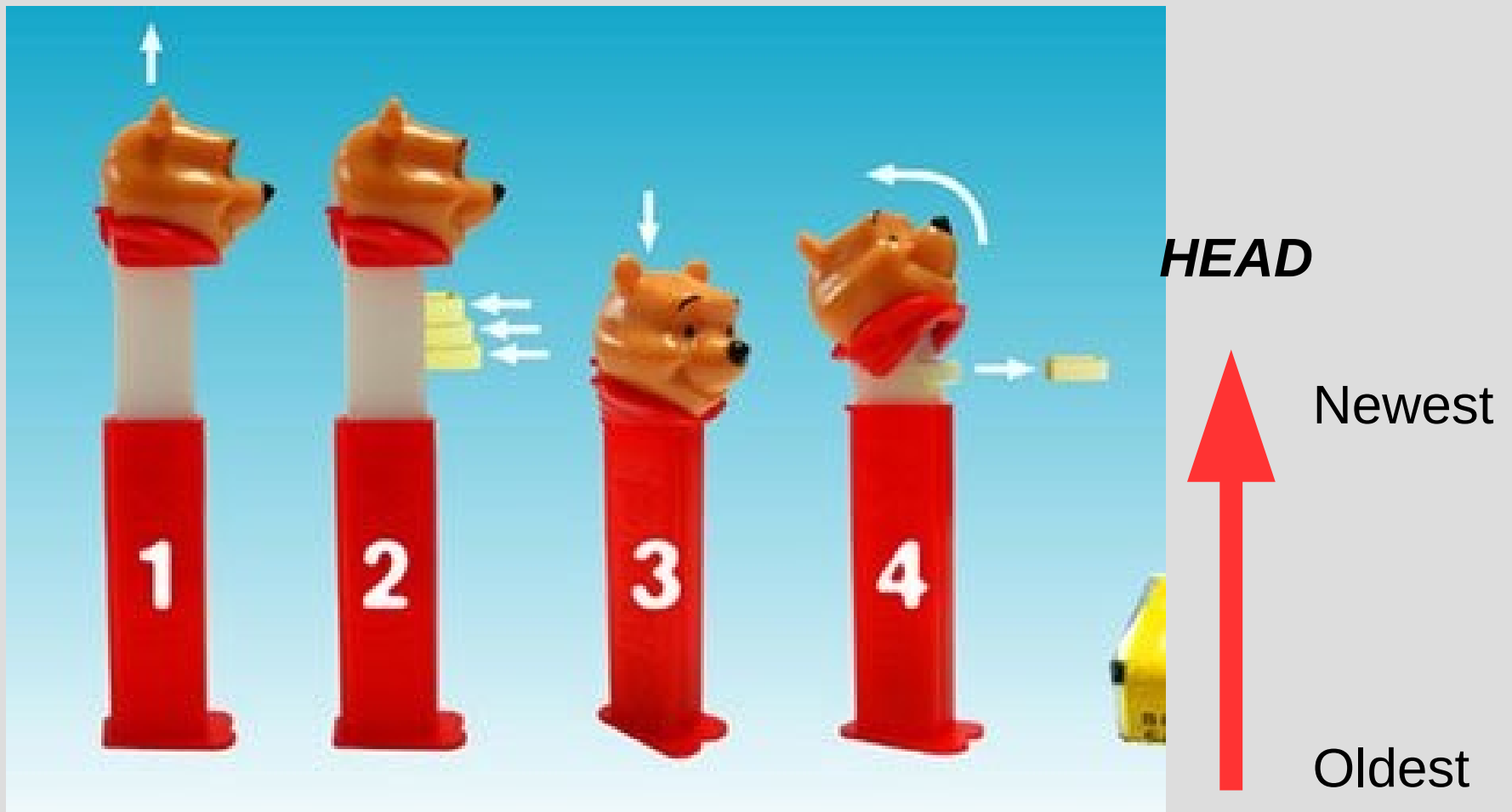
```
$git log
```

- Top entry (most recent ***commit***) is the ***HEAD***

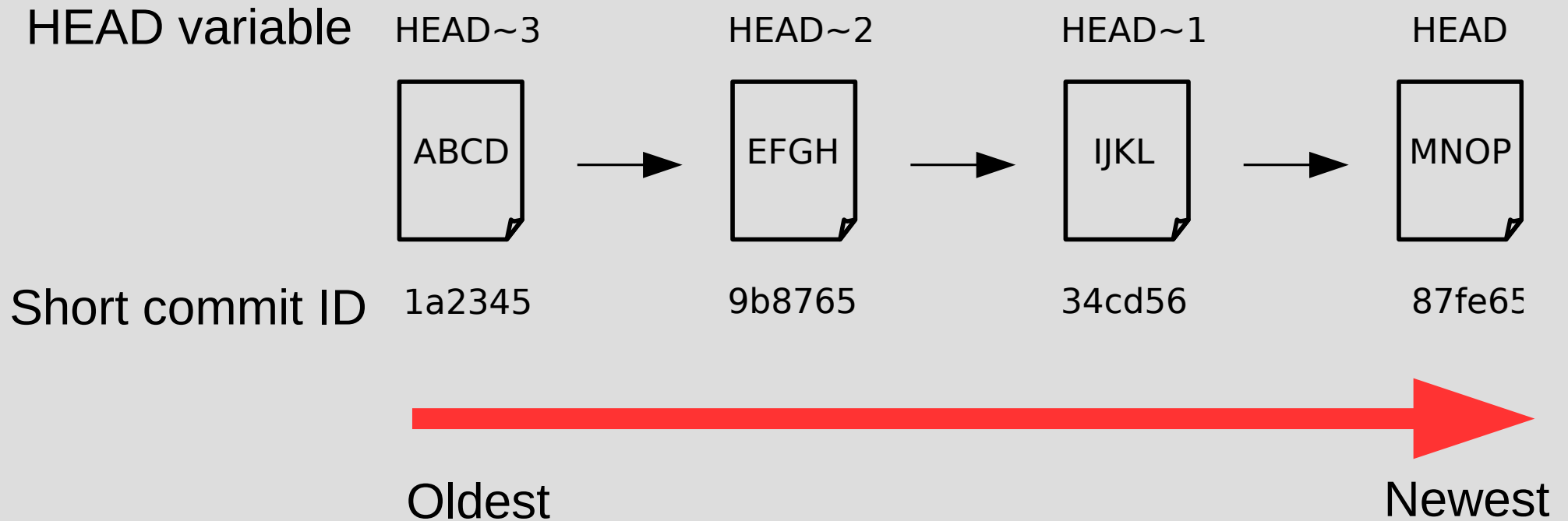


HEAD: Most recent ***commit***

Handling History



Handling History



Handling History

- Compare old versions to current

```
$git diff HEAD~2 paper.txt
```

Handling History

- Compare old versions to current

```
$git diff HEAD~2 paper.txt
```

```
$git diff f705802 paper.txt
```

Handling History

- Compare old versions to current

```
$git diff HEAD~2 paper.txt
```

```
$git diff f705802 paper.txt
```

```
$git diff f705802e2ff63c91b39599a866c1628d7a6b0ae6\  
paper.txt
```

Use whichever method you prefer (probably the last one, right?).

Handling History

- Return to a previous commit (version) of our document

```
$git checkout HEAD~2 paper.txt
```

Handling History

- Look at our log; what's different?

Handling History

- Look at our log; what's different?
- What about our file?

Handling History

- Look at our log; what's different?
- What about our file?
- Open file, make some changes; ***add*** and ***commit***

Handling History

- Look at our log; what's different?
- What about our file?
- Open file, make some changes; ***add*** and ***commit***
- Look at our log

Handling History

- Look at our log; what's different?
- What about our file?
- Open file, make some changes; ***add*** and ***commit***
- Look at our log

We did it!



Handling History: Take Home

View *log* of commits



Checkout HEAD~n



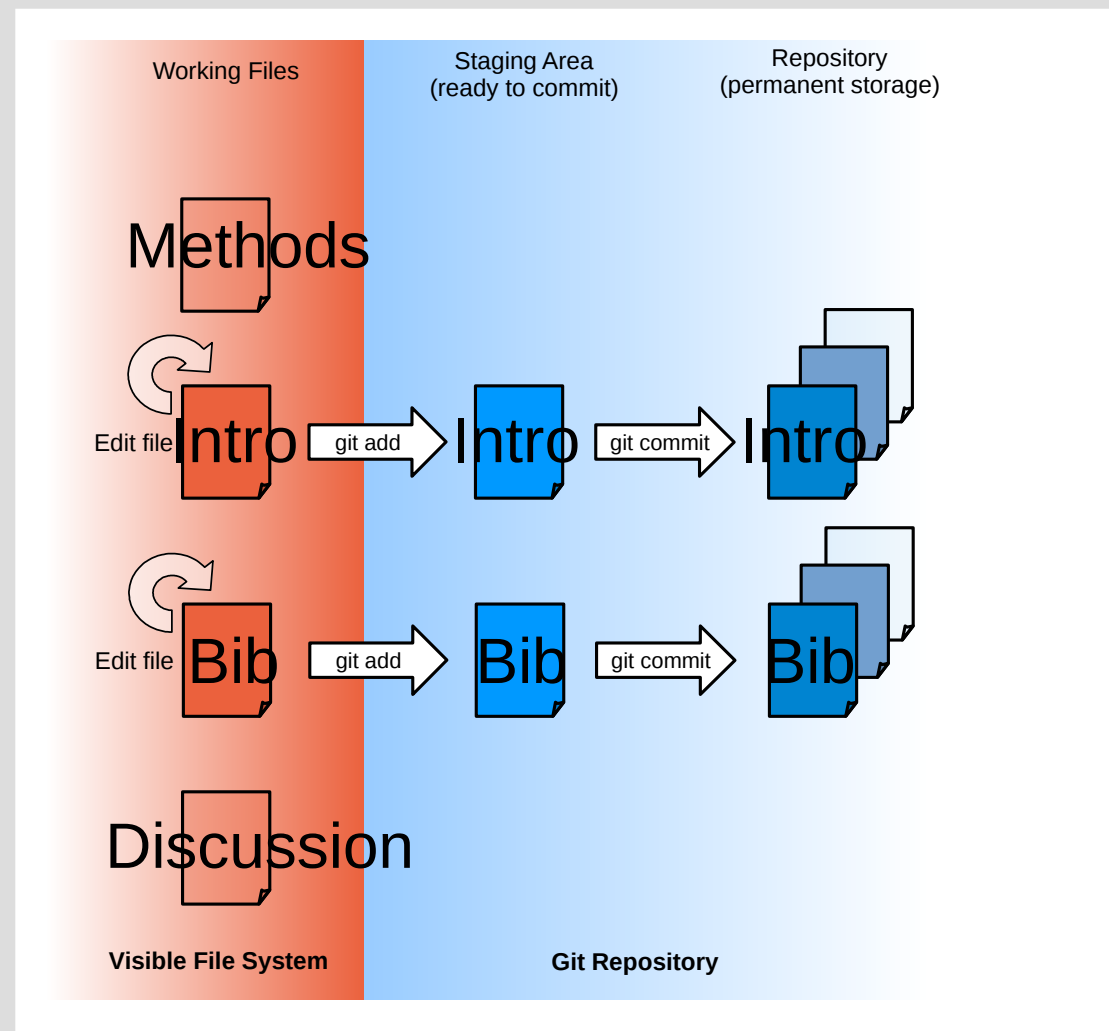
Add & Commit new changes

Project Organization

- “Commit early, commit often” - Someone on Twitter

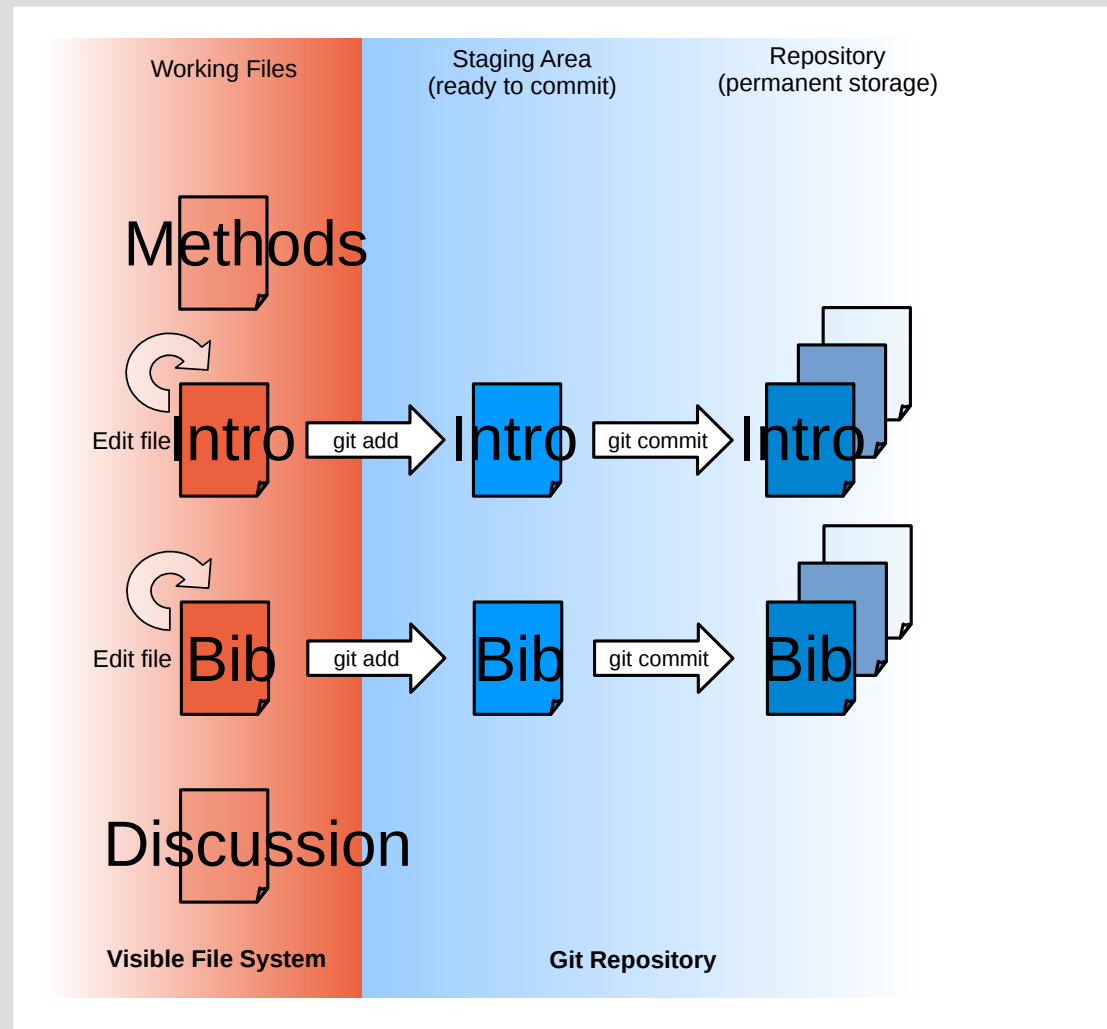
Project Organization

- “Commit early, commit often” - Someone on Twitter
- Project segmentation



Project Organization

- “Commit early, commit often” - Someone on Twitter
- Project segmentation
- Ignore files



Project Organization

Ignore files

- Create some additional files, check ***status***

Project Organization

Ignore files

- Create some additional files, check ***status***
- Create a file called: `.gitignore`
 - Add name(s) of files to `.gitignore` file

Graphical User Interfaces (**GUI**)



GUI – pronounced “gooey”