IT Club | UXCAM

GIT WORKSHOP

Day 1: Introduction to Git and VCS

What’s in this lecture?

What's in this lecture?

IT Club | uxcam

Part 1 (Basic Theory)
Version Control System
Why Git

Part 2 (Working with Git)
Setting up a git repository
Adding and committing files
Status
Logs

Let’s get started!

Part 1

Version Control System and Git

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.

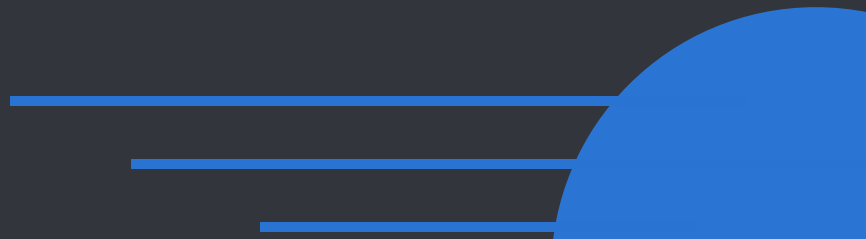
-https://git-scm.com

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.

-https://git-scm.com

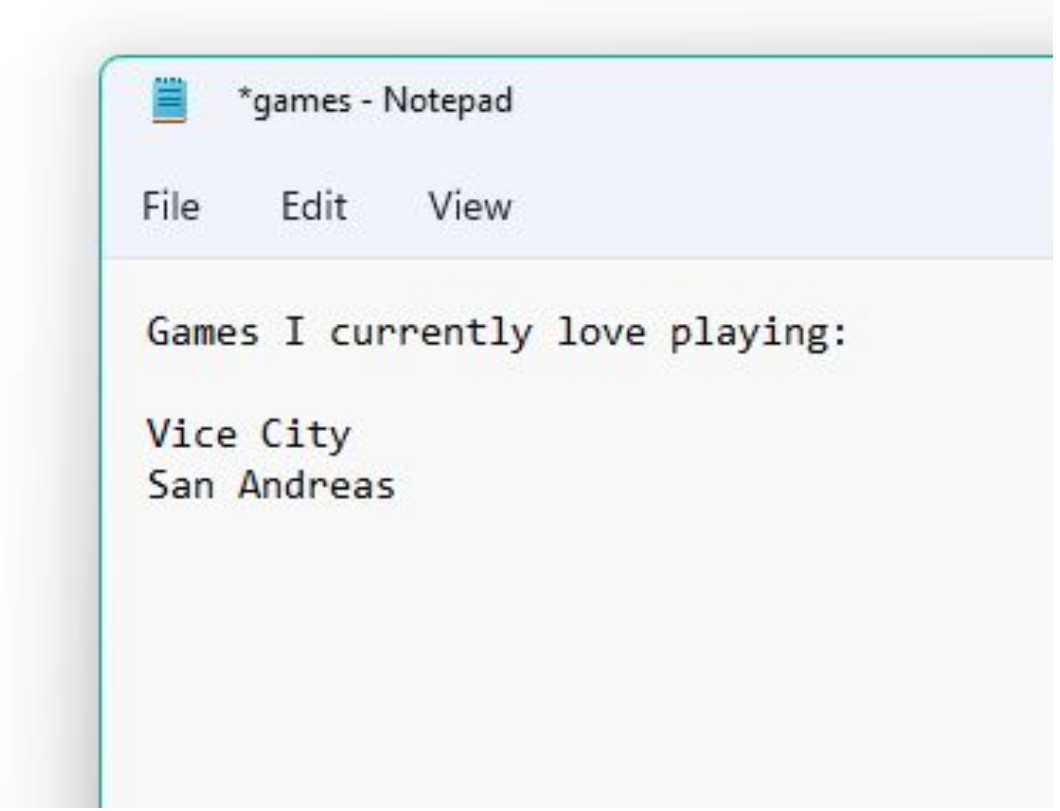
Version Control System?



Say I want to create a list of games that I love playing!

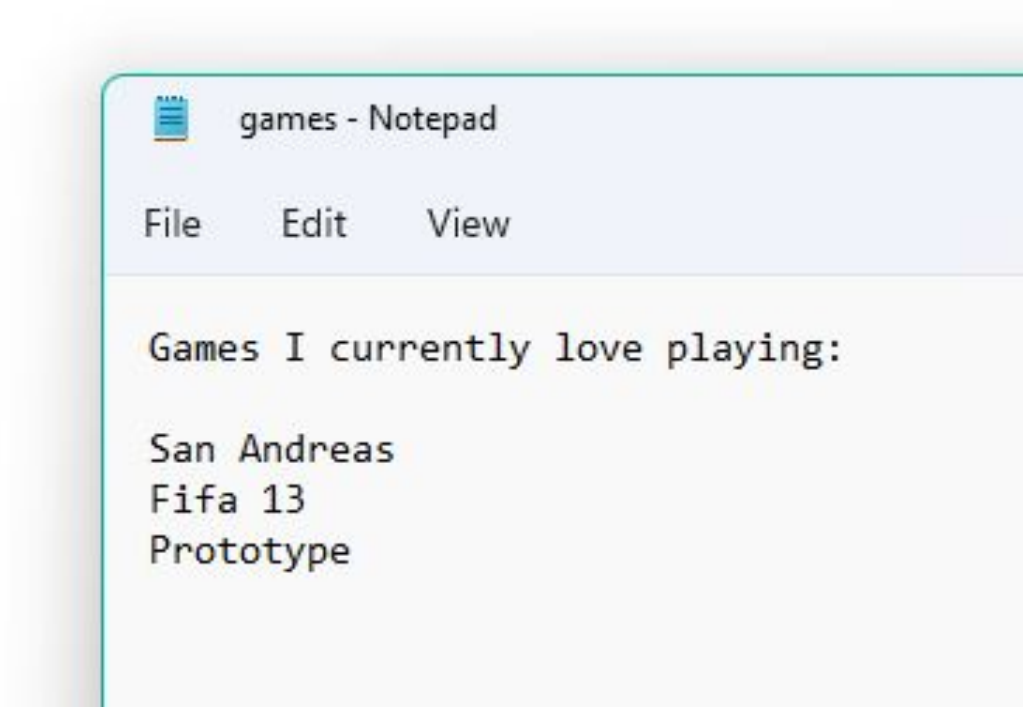
2012


games


*games - Notepad
File
Edit
View
Games I currently love playing:
Vice City
San Andreas

2013


games


games - Notepad
File
Edit
View
Games I currently love playing:
San Andreas
Fifa 13
Prototype

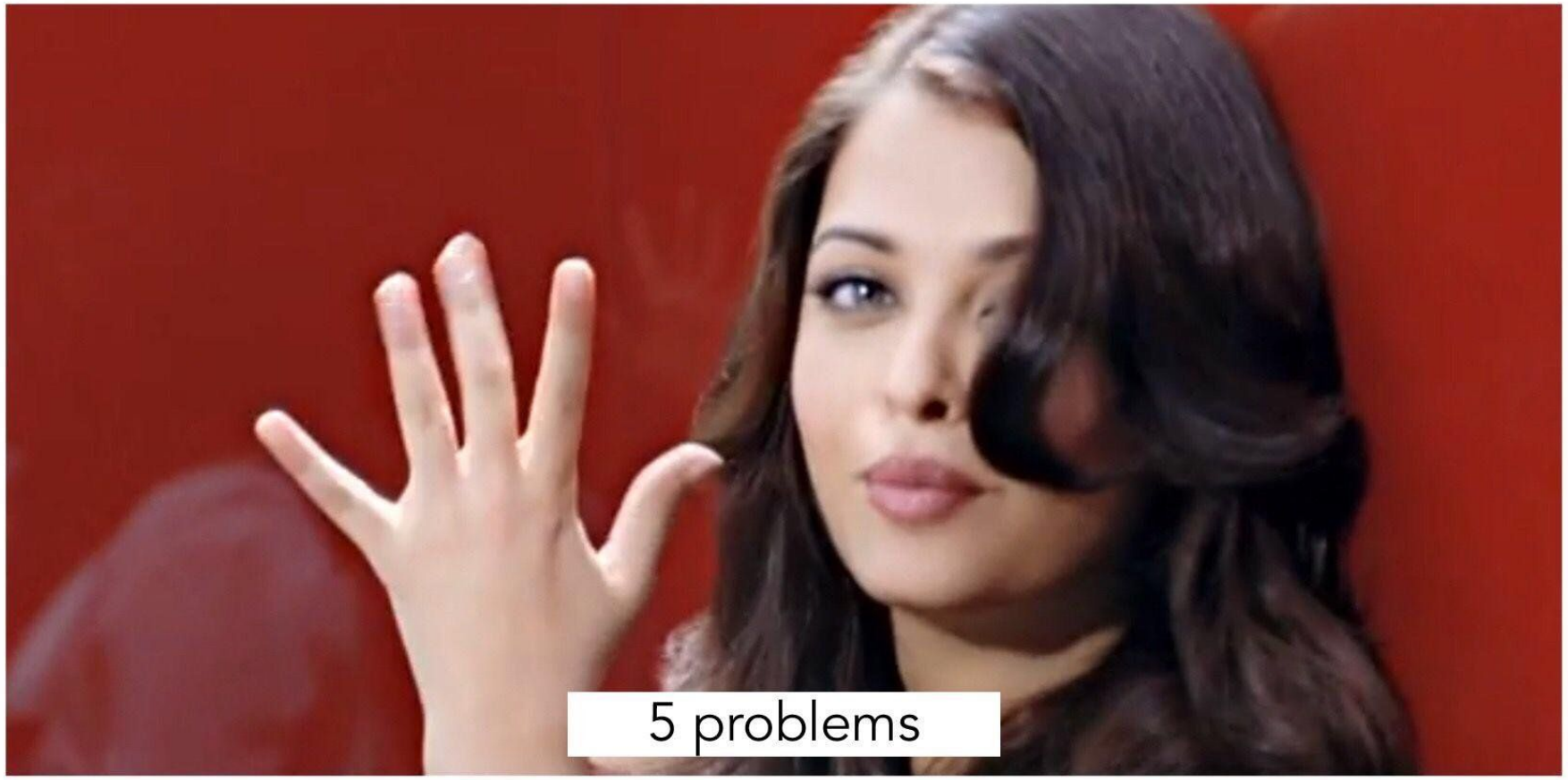
2014

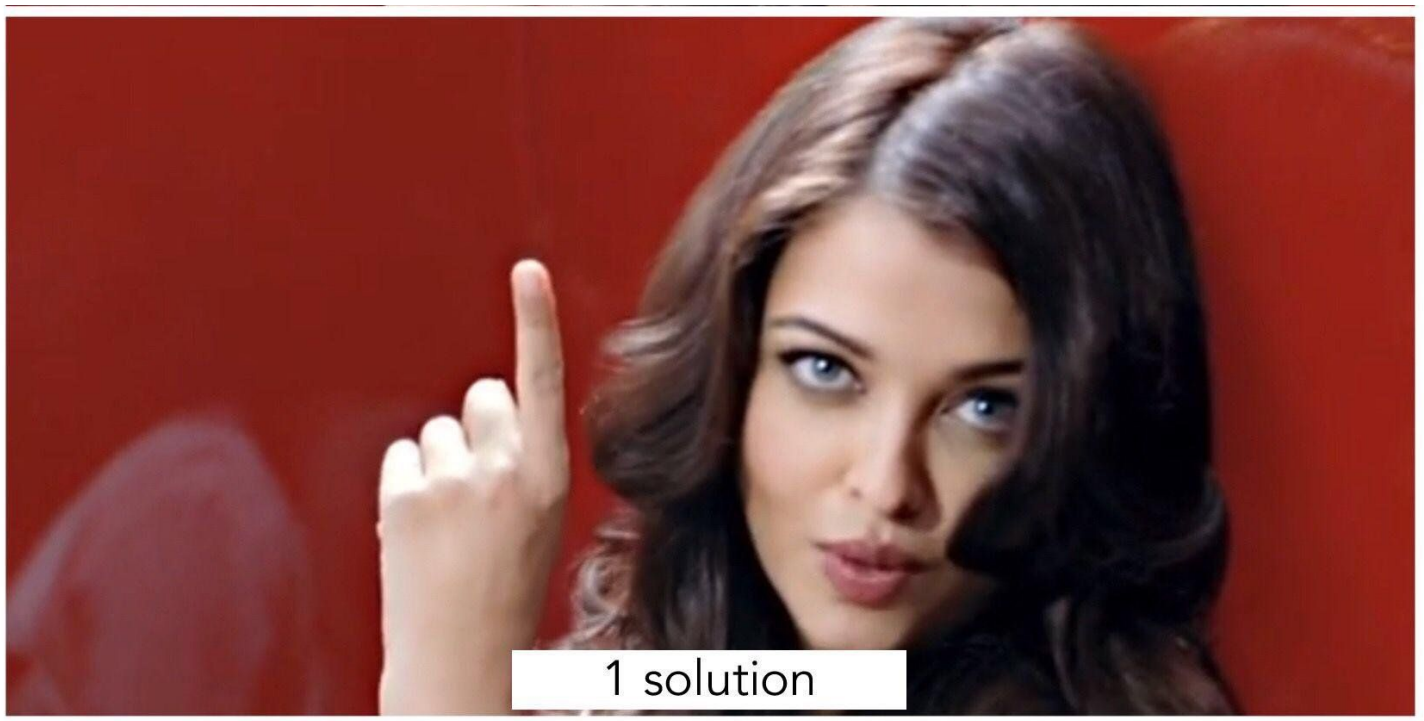

games


*games - Notepad
File
Edit
View
Games I currently love playing:
Prototype
GTA V

Problems with our model:

1. **When** were the changes made?
2. **Why** were the changes made?
3. How do I add **collaborators**?
4. **Who** made the changes?
5. How do I get back the **old list**?


5 problems



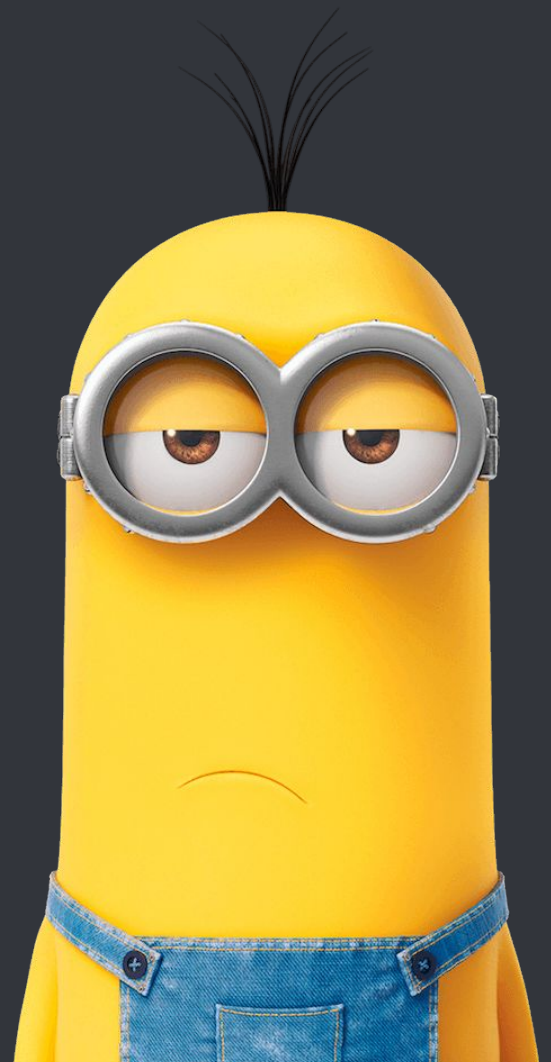
Version Control System!

Version control systems (VCSs) are tools used to track changes to source code (or other collections of files and folders).

VCSs track changes to a folder and its contents in a series of snapshots, where each snapshot encapsulates the entire state of files/folders within a top-level directory.

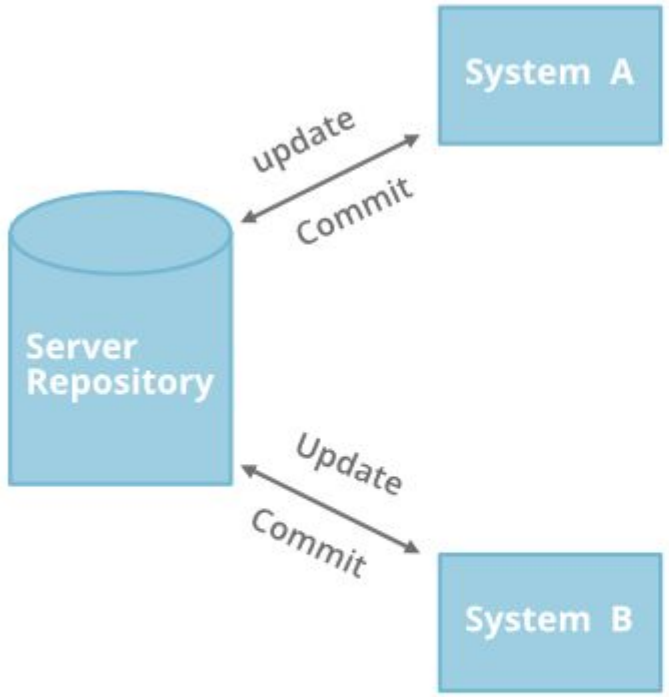
Features:

- Help maintain a history of changes
- Facilitate collaboration
- Maintain metadata like who created each snapshot, messages associated with each snapshot, and so on.

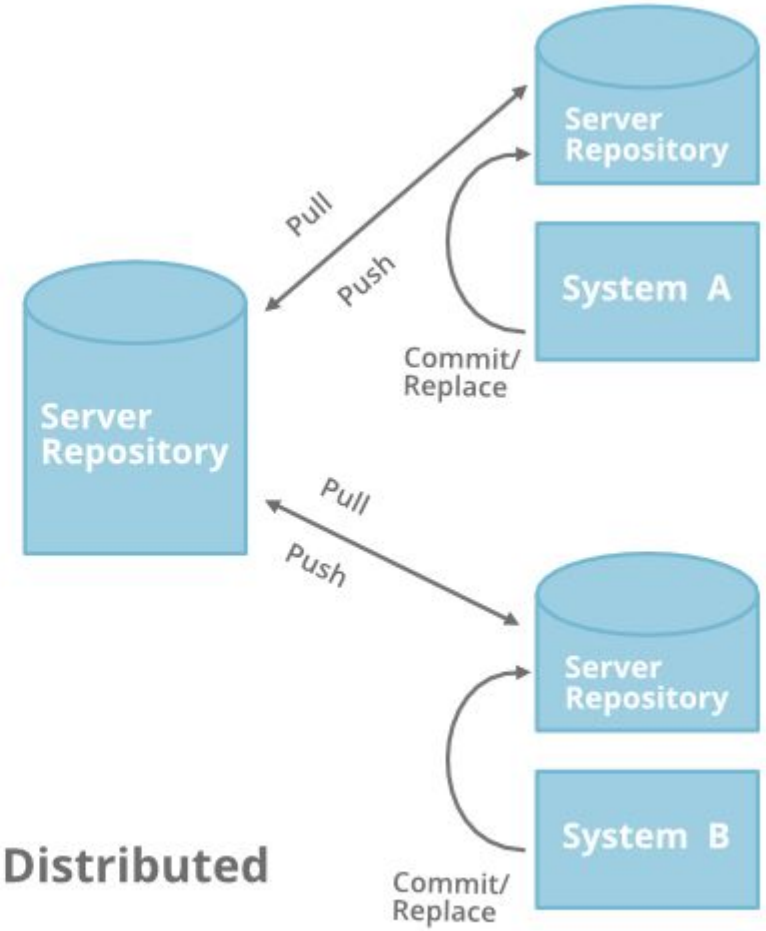


Distributed?

Distributed VCS



Centralized



Distributed

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.

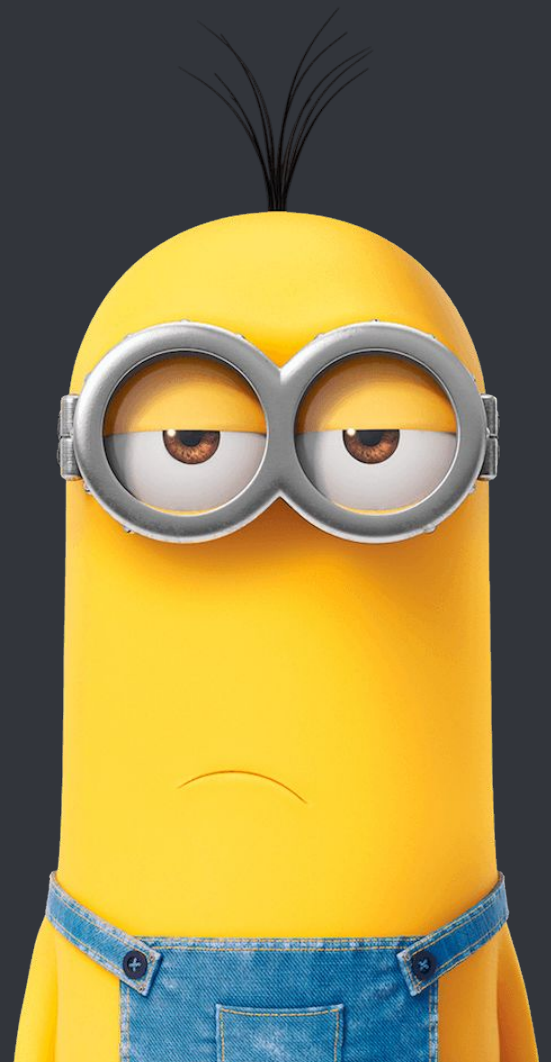
-https://git-scm.com

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.

-https://git-scm.com

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.

-https://git-scm.com



Why Git?


CVS
CONCURRENT
VERSIONS
SYSTEM


SUBVERSION

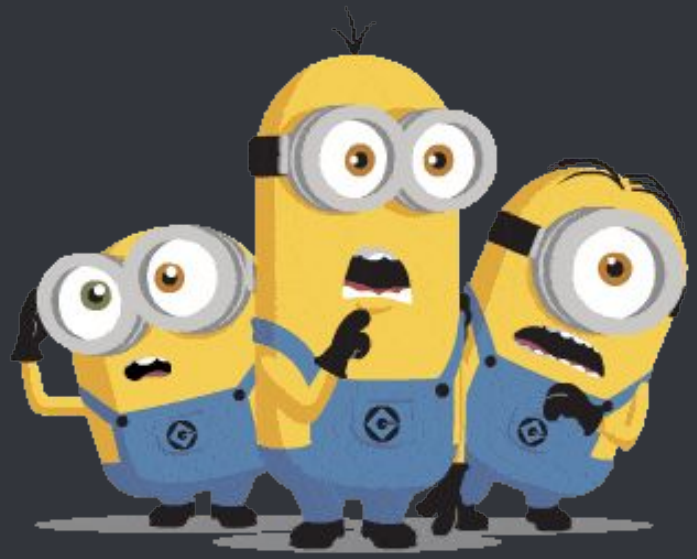

git

Why git?

IT Club | uxcam

- Free and open source.
- Works on different range of OS
- Large community, good availability of documentation and resources.

Questions?



Part 2

Working with Git

Verifying git installation

Verifying git installation


```
git config --global user.name "Rick Astley"

git config --global user.email "hello@rick.com"
```

Let’s set up our first repo!
(Demo time)

Let’s break it down!

Breakdown

```
git init
```

It creates an empty git repository in the working directory.

A hidden folder named .git is created which contains all the information of the repository such as commit history, etc.

Breakdown

IT Club | uxcam

```
git add <files>
```

Adds files to a so called “staging area”

These are the files git keeps track of.

```
git commit -m “Commit Message”
```

Takes a snapshot of the project’s current state.

Can think of each snapshot as a “checkpoint”. You can revert back here later as you wish.

Let’s make some changes

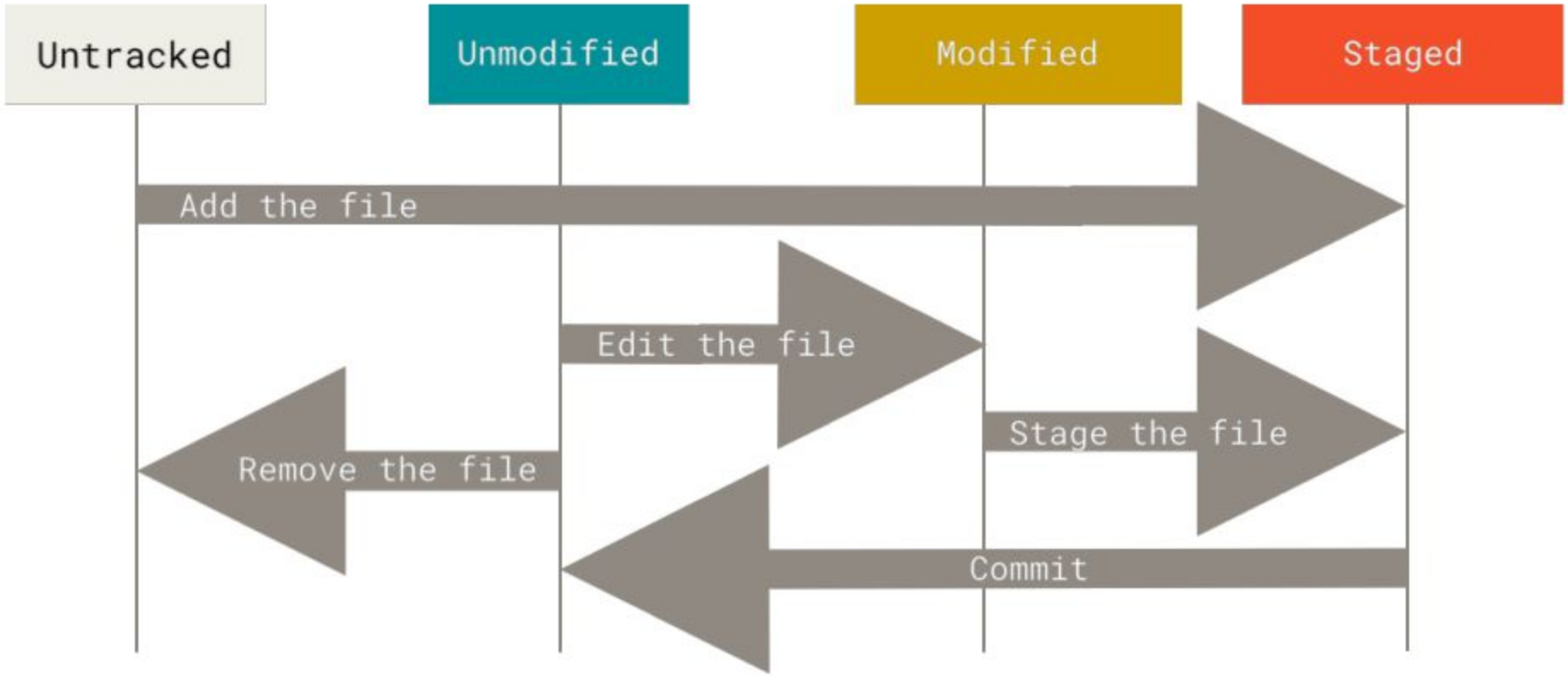
Breakdown

IT Club | uxcam

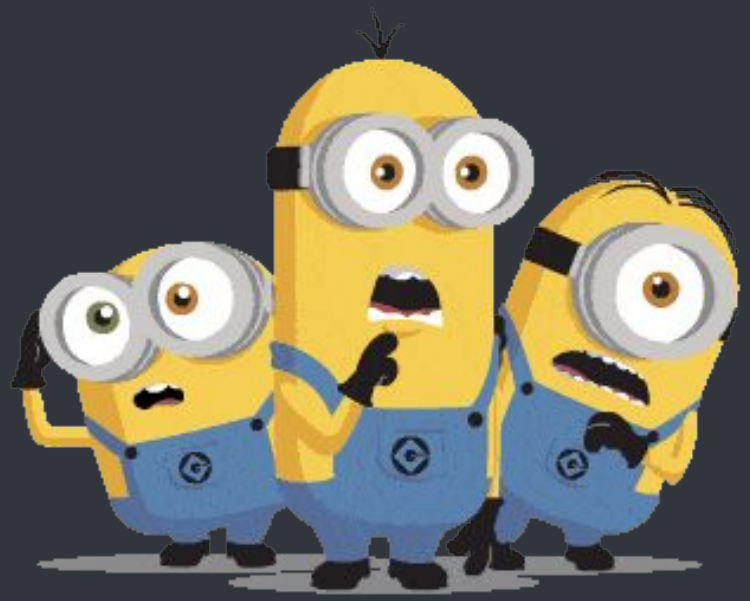
Now that we've made some changes to the project, let's check the repo's status.

```
$ git status
```

Breakdown


Untracked
Unmodified
Modified
Staged
Add the file
Edit the file
Stage the file
Remove the file
Commit

Questions?



Misc Stuff

There may be files which you don't want git to track.

You can ask git to ignore those files by listing them out in a file named .gitignore which resides in your working directory

Use cases:

Sensitive info (environment variables)

Modules

log

```
$ git log
```

Shows the commit log.

```
$ git log --all -- graph --decorate
```

```
$ git diff
```

Shows changes between different data sources.

By default, shows the difference between the staging area and the working directory

```
$ git config alias.[short_name] <command>
```

assign shorter name to commands

Questions?