

## VERSION CONTROL WITH

---



Some of the material comes  
from the "CodeRefinery" project



@JacquesDainat



@jacquesdainat@genomic.social



Juke34



Jacques.dainat@ird.fr

Jacques Dainat Ph.D.



Reproducibility is a fundamental pillar of the scientific method!

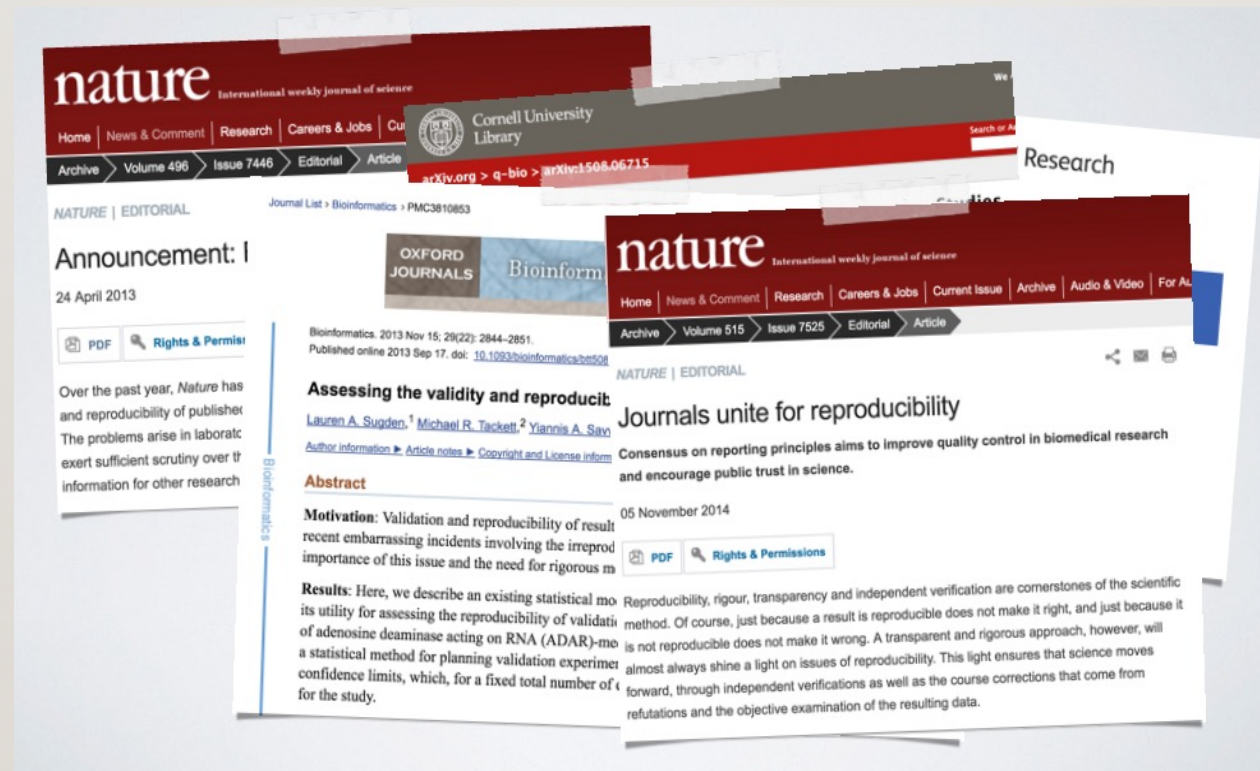


Why should we talk about reproducible research when research should in essence be “reproducible”?

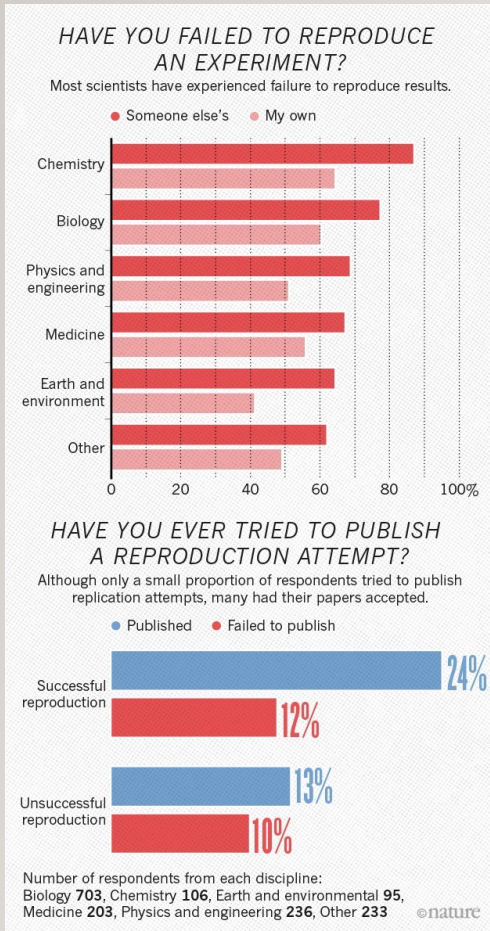
Crisis discovered middle 2000s

Getting bigger early 2010s

Many publications tackle the problem





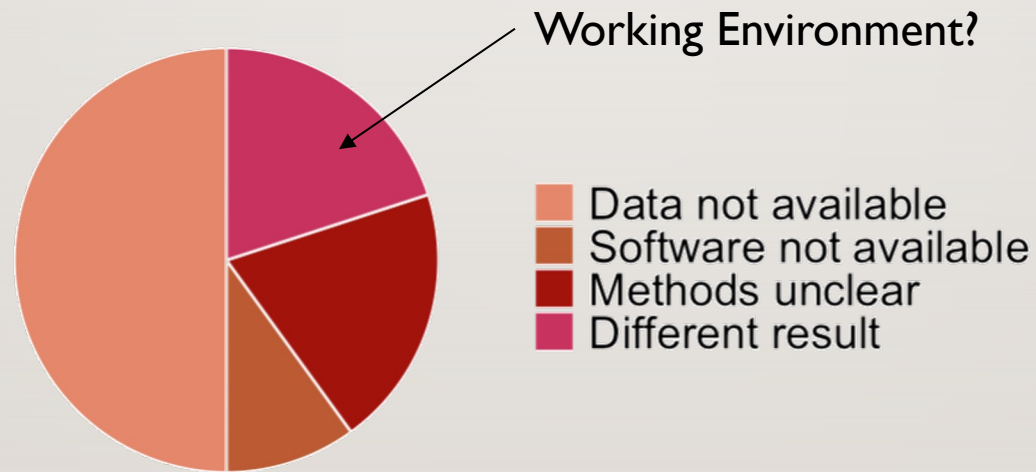


## Nature 2016 (<https://doi.org/10.1038/533452a>) 1,500 scientists lift the lid on reproducibility

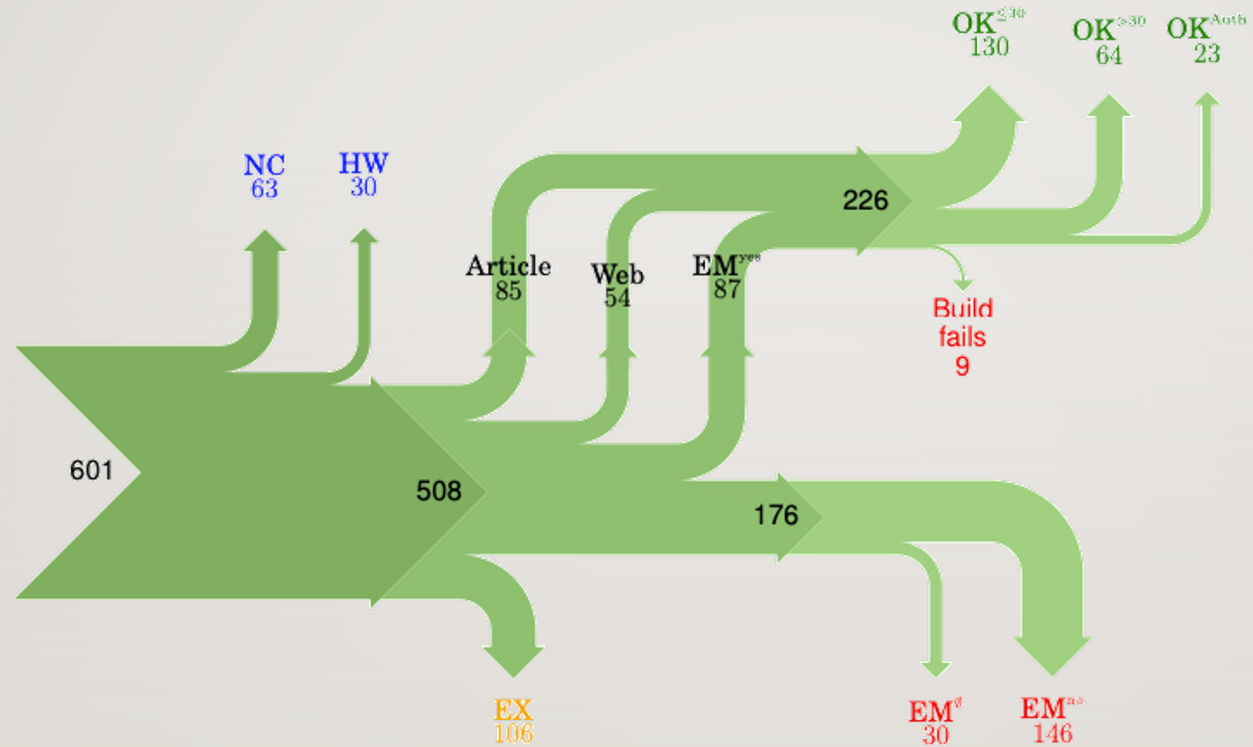
- Irreproducible experiments are a problem across all domains of science.
- >70% of researchers have tried and failed to reproduce another scientist's experiments
- More than half have failed to reproduce their own experiments.

## Replication of data analyses in 18 articles on microarray-based gene expression profiling published in Nature Genetics in 2005–2006:

Adopted from Ioannidis et al. "Repeatability of published microarray gene expression analyses", Nature Genetics 41 (2009) doi:10.1038/ng.295



computer science

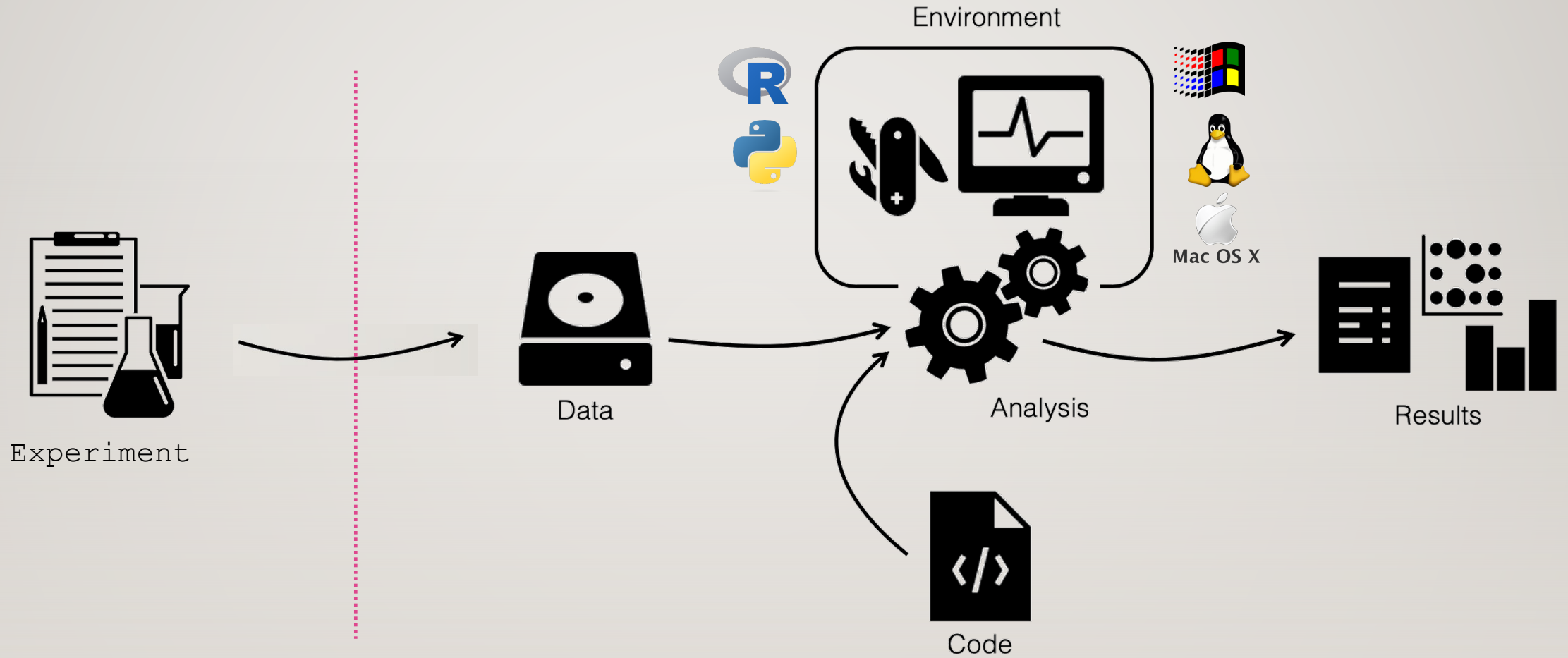


They succeeded to compile only **43%** of the codes

Working Environment?



# Reproducibility Crisis



## Tools for reproducible research

 **git** • Track changes and versions (with VCS)

 **conda**

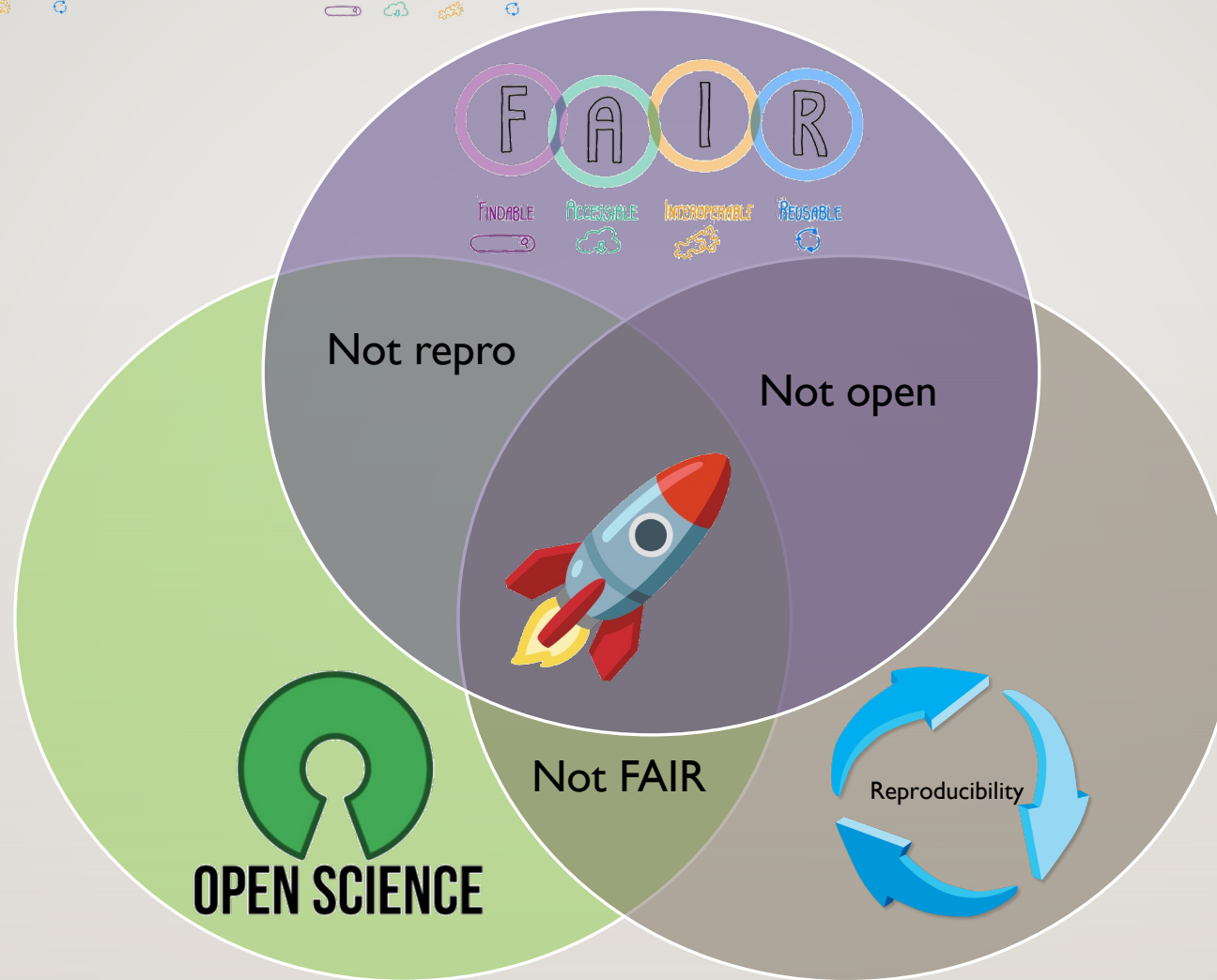


• Dependencies hell (scripts, tools, DB, ref etc)

 **nextflow**  
 **snake make**

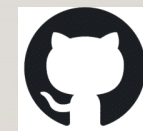
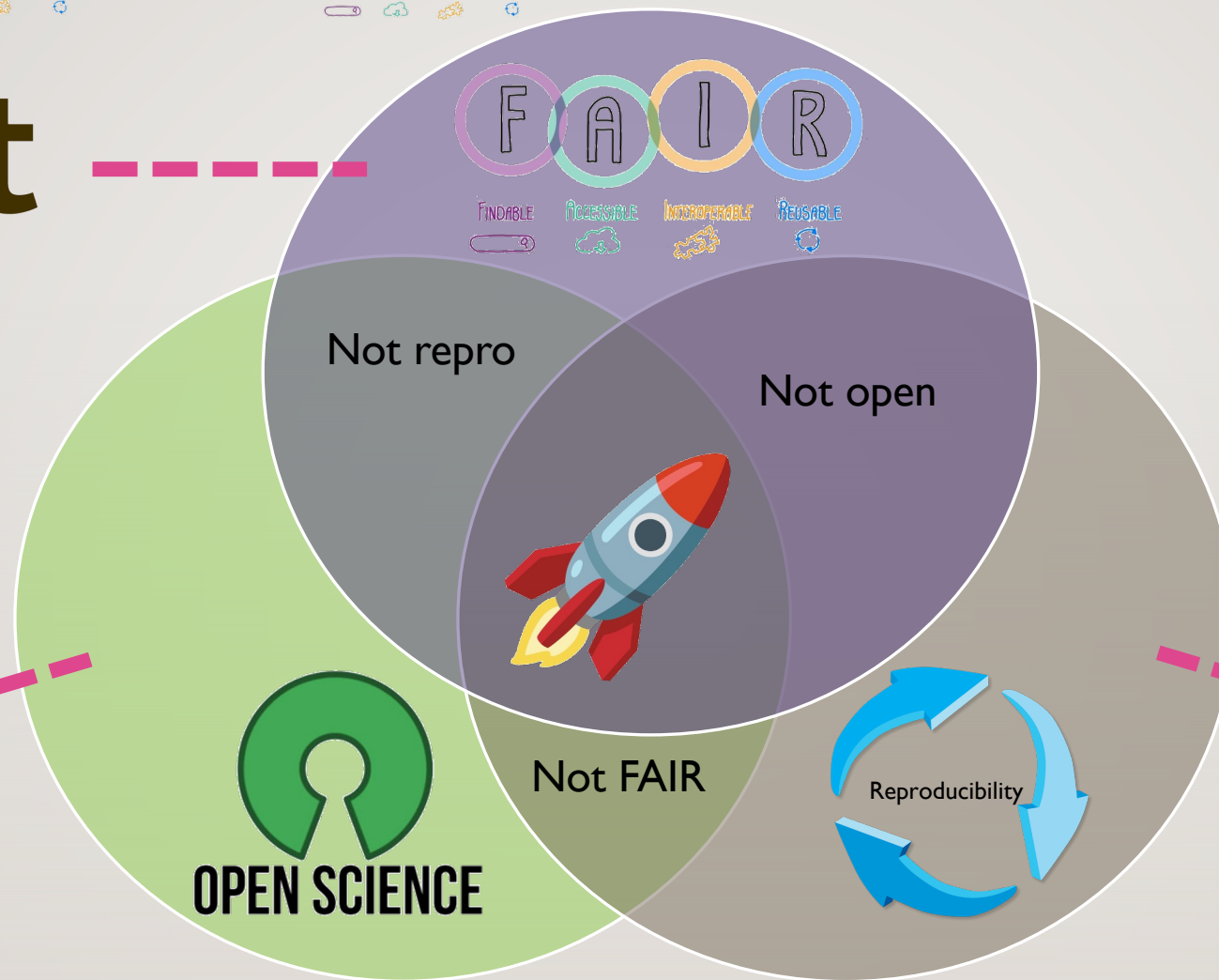
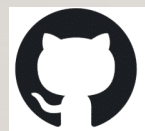
• Workflow manager system







git



\* Allows you to (but you don't have to)



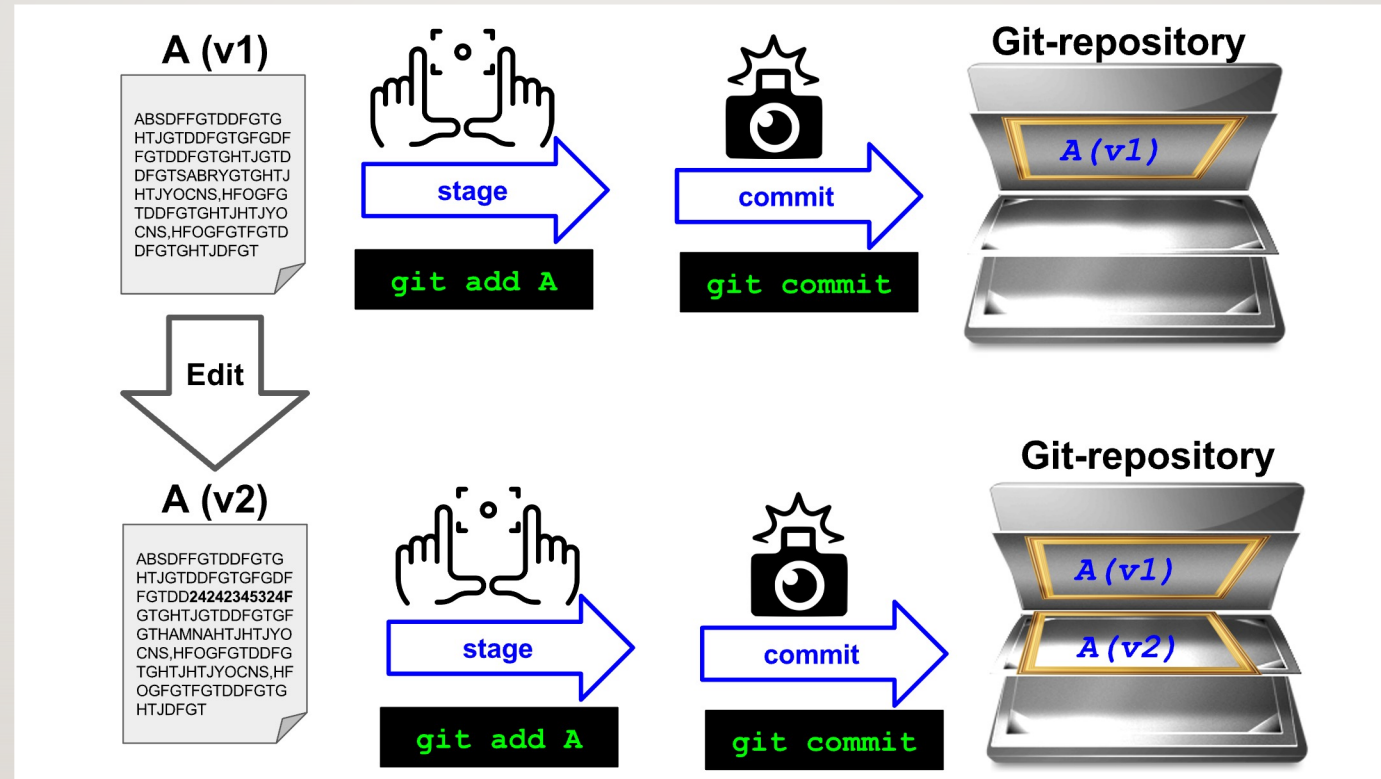
Linus Torvalds

- Monitoring changes
- Collaboration

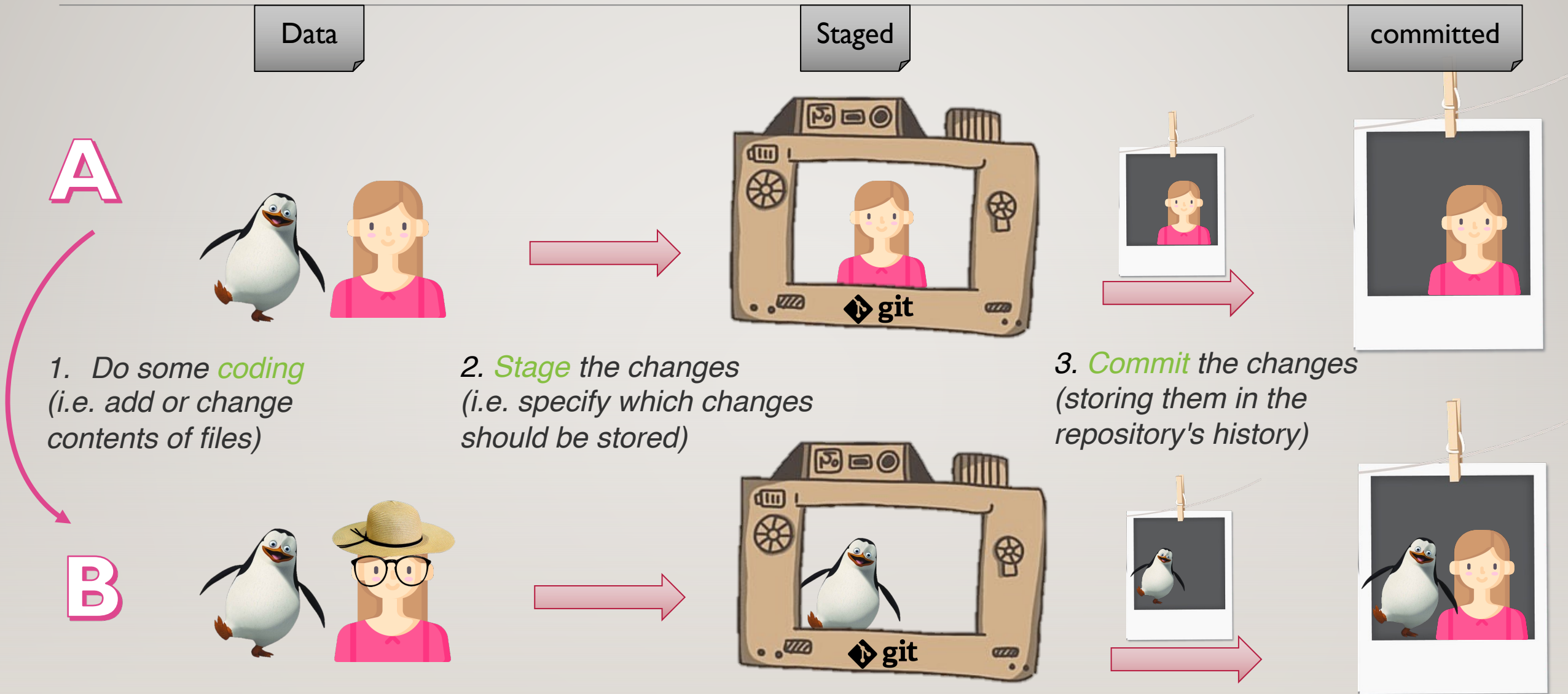




- System which **records snapshots** of a project

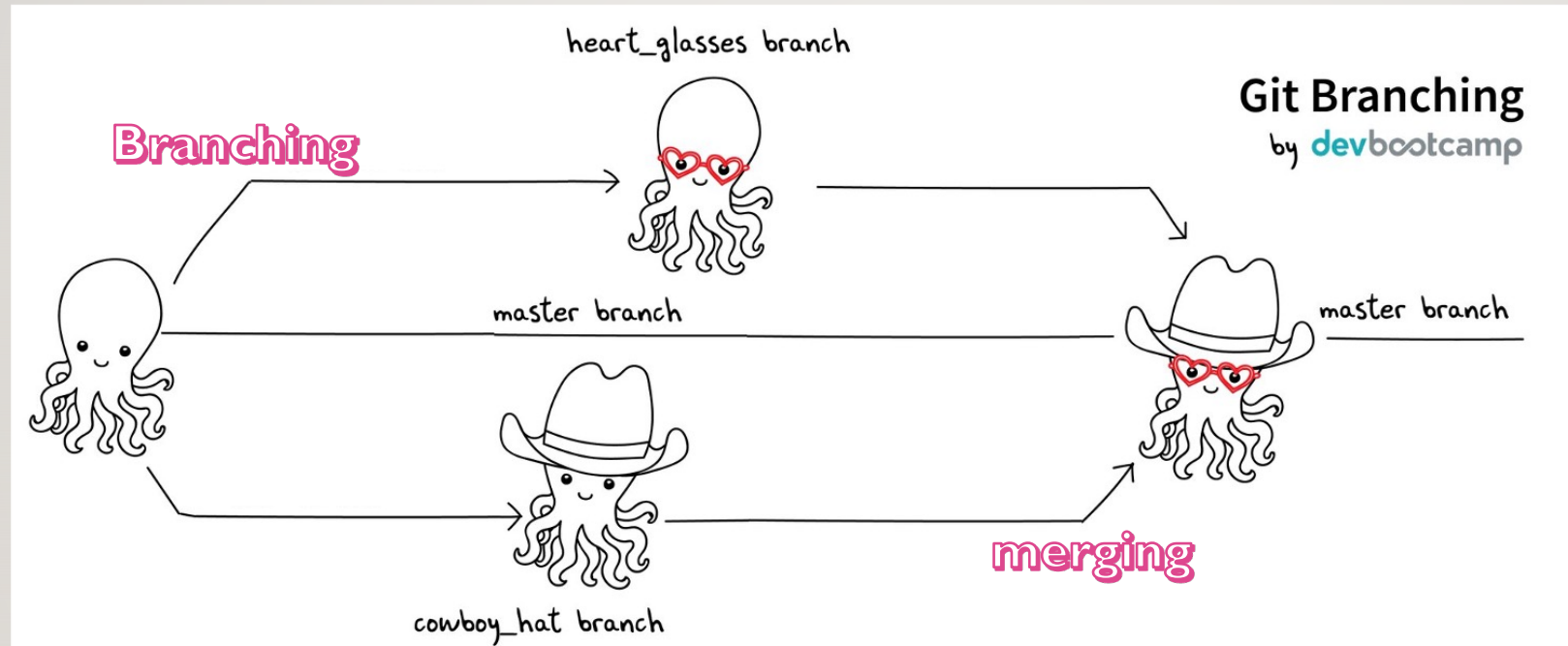


# Staged vs commit



- Implements **branching**:

- You can work on several feature branches and switch between them
- Different people can work on the same code/project without interfering
- You can experiment with an idea and discard it if it turns out to be a bad idea



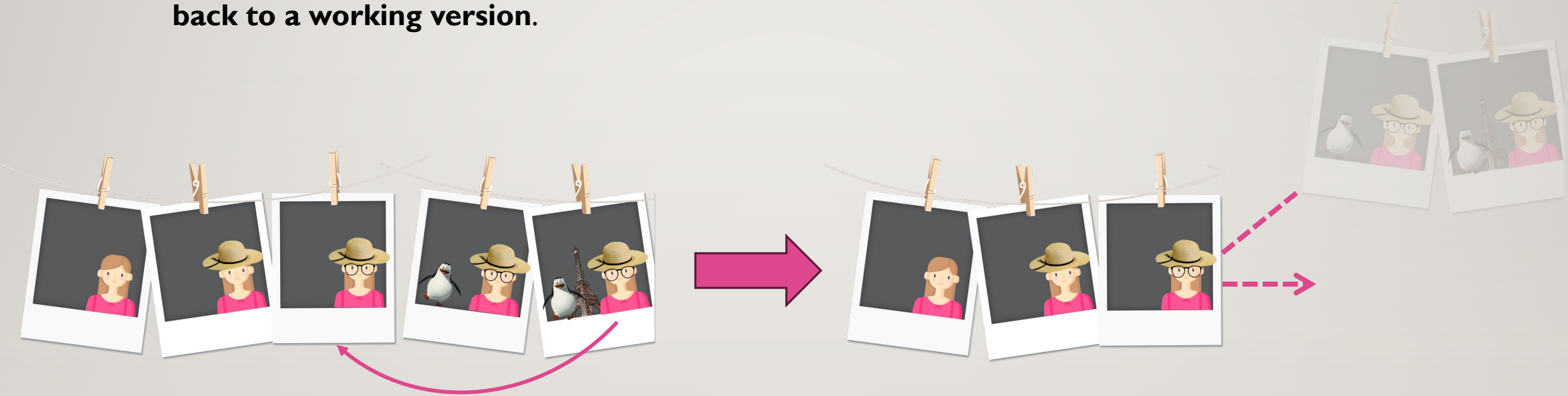
- Implements **merging**:

- Person A and B's simultaneous work can be easily combined



- **Roll-back functionality**

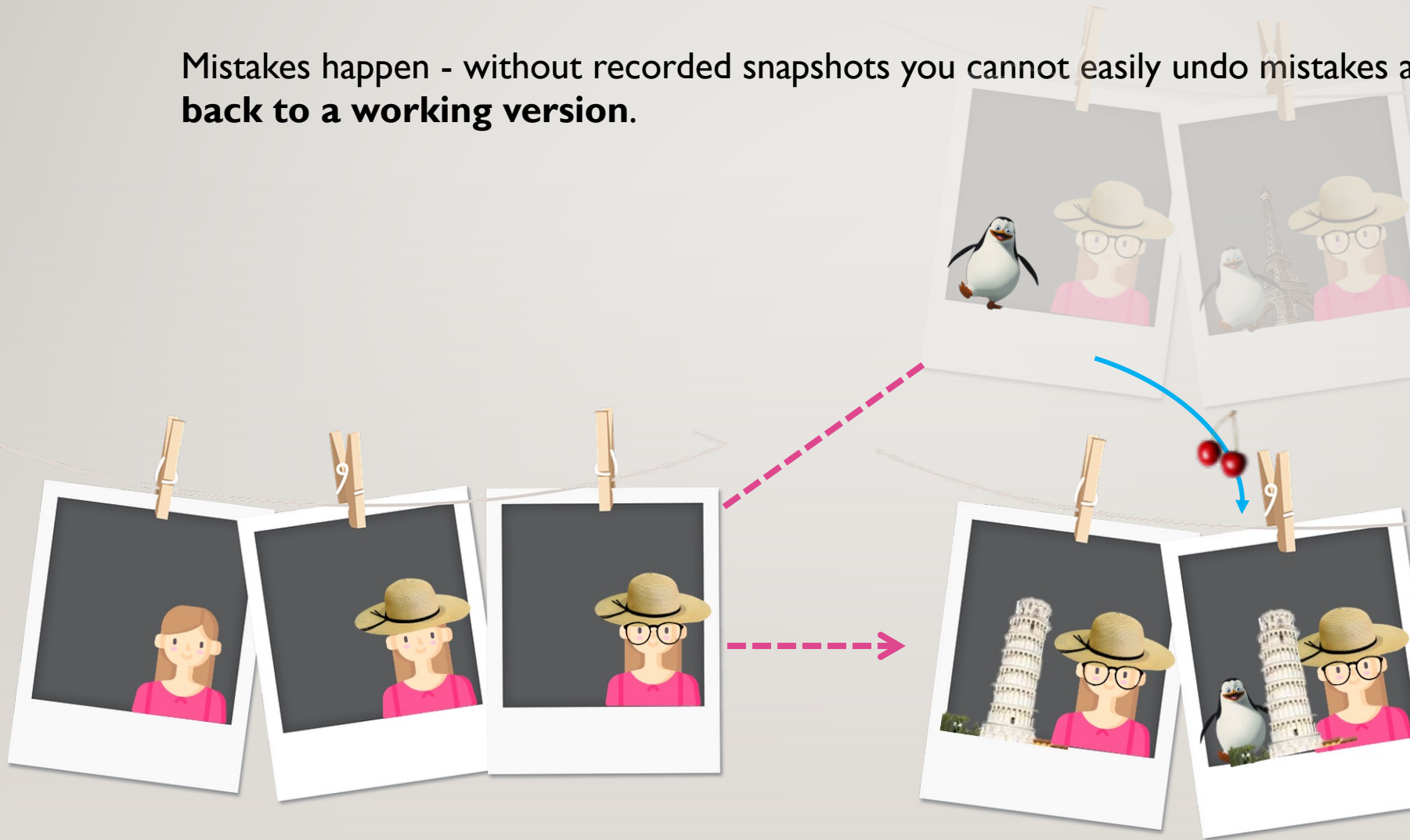
Mistakes happen - without recorded snapshots you cannot easily undo mistakes and **go back to a working version.**



*What has been committed at each step here?*

- Cherry picking

Mistakes happen - without recorded snapshots you cannot easily undo mistakes and **go back to a working version.**



- **Software** (this is how it started but Git/GitHub can track a lot more)
- **Scripts**
- **Documents** (plain text files much better suitable than Word documents)
- **Manuscripts** (Git is great for collaborating/sharing LaTeX or Quarto manuscripts)
- **Configuration files**
- **Website sources**
- **Data** (!\ with huge data)



## Reproducibility

- How do you indicate which version of your code you have used in your paper?
- When you find a bug, how do you know **when precisely** this bug was introduced (Are published results affected? Do you need to inform collaborators or users of your code?).

### Collaboration

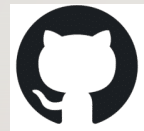
With version control, none of these are needed anymore (or have much simpler answers):

- *“I will just finish my work and then you can start with your changes.”*
- *“Can you please send me the latest version?”*
- *“You never got the code I send by email? Maybe the spam filter marked it as malicious?”*
- *“Where is the latest version?”*
- *“Which version are you using?”*
- *“Which version have the authors used in the paper I am trying to reproduce?”*

VCS => To keep track of changes and versions

Subversion / Mercurial / **Git** => One of the most popular VCS in use today

Bitbucket / GitHub / GitLab => hosting services for Git repositories



```
> git log
```

```
commit 9fc68014300136d412f350d25309e3c209dbeeb3
Author: Jacques Dainat <jacques.dainat@gmail.com>
Date:   Fri Mar 3 17:50:41 2023 +0100
```

```
    add github usernames
```

```
commit 1b399d4f42b9700af3daa07d4d9f511b43d5b701
Author: jhayer <juliette.hayer@gmail.com>
Date:   Tue Jan 17 15:38:27 2023 +0100
```

```
    fixed condition to run krakentools extract
```

Unique  
identifier

Who

When

What

One  
commit

One  
commit



Differences ?

```
commit 9fc68014300136d412f350d25309e3c209dbeeb3
Author: Jacques Dainat <jacques.dainat@gmail.com>
Date:   Fri Mar 3 17:50:41 2023 +0100

    add github usernames

commit 1b399d4f42b9700af3daa07d4d9f511b43d5b701
Author: jhayer <juliette.hayer@gmail.com>
Date:   Tue Jan 17 15:38:27 2023 +0100

    fixed condition to run krakentools extract
```

```
> git diff 9fc68014300136d412f350d25309e3c209dbeeb3 1b399d4f42b9700af3daa07d4d9f511b43d5b701
```

File

```
--- a/README.md
+++ b/README.md
@@ -38,5 +38,5 @@ d

  ## Author and contributors

- Juliette Hayer
- Jacques Dainat
+ Juliette Hayer (@jhayer)
+ Jacques Dainat (@Juke34)
```

Changes

Using git blame, you can see the entire git revision of the file with respect to each line.

Git blame will come in handy when you want to know which commit is responsible for the changes and which author has committed those changes

```
> git blame myfile.nf
```

9797e097	(Jacques Dainat	2023-01-23	17:08:50	+0100	17)	* [Usage] (#usage)
9797e097	(Jacques Dainat	2023-01-23	17:08:50	+0100	18)	* [Parameters] (#parameters)
4179de43	(jhayer	2023-02-17	16:12:13	+0100	19)	* [Update] (#update)
4179de43	(jhayer	2023-02-17	16:12:13	+0100	20)	* [Uninstall] (#uninstall)

commit                      Who                      When                      line                      Content of the file

Despite the benefits, let's be honest, there are some difficulties:

- One more thing to learn (it's probably worth it and will save you more time in the long run; basic career skill).
- Difficult if some people don't want to use it (in the worst case, you can version control on your side and send them versions).
- Advanced things can be difficult, a bit too many gotchas (basics are often enough, ask others for help when needed).





Let us explore a famous **existing Git repository** on GitHub, which was used to produce the Event Horizon Telescope images:

<https://github.com/achael/eht-imaging>

The goal here is not to teach GitHub yet but rather to get a glimpse of the wider picture.

- \* How many commits?
- \* How many releases?
- \* How many Forks/Stars?
- \* How many contributors?
- \* Documentation (Readme/Readthedocs)



Let us see the social aspect :

Simple markdown:

<https://github.com/Juke34>

Advanced website / blog / CV:

<https://astrobiomike.github.io>



### The Hooks:

- Documentation
- Code coverage
- Integration Continue (CI)
- DOI
- External hook (e.g. bioconda)
- etc

<https://github.com/NBISweden/AGAT>



# *The End*

***Thank you for your attention***