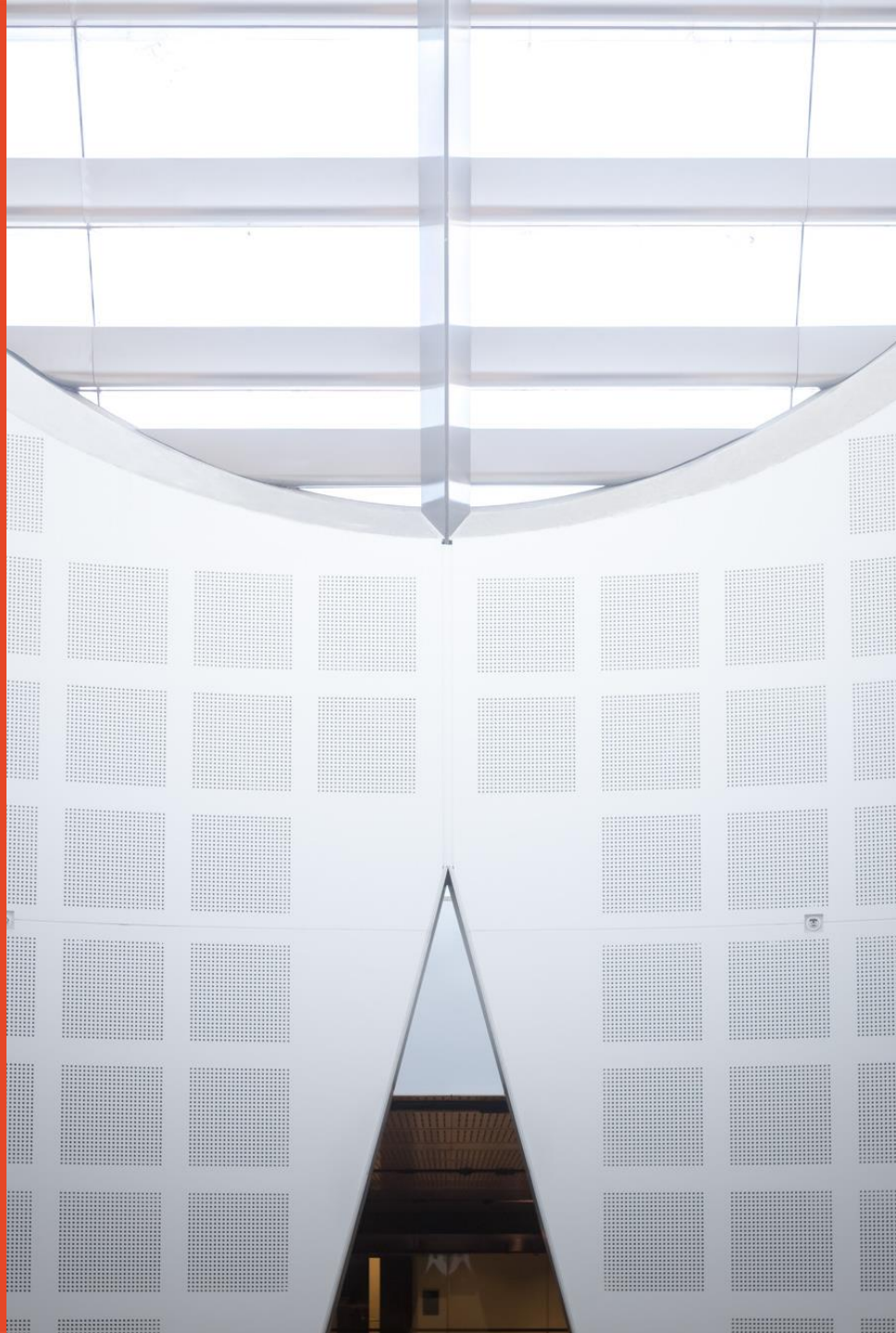
INFO1111: Computing 1A Professionalism

Week 5:
- Finding Information
- Collaboration
- Git

Professor David Lowe

School of Computer Science

THE UNIVERSITY OF SYDNEY



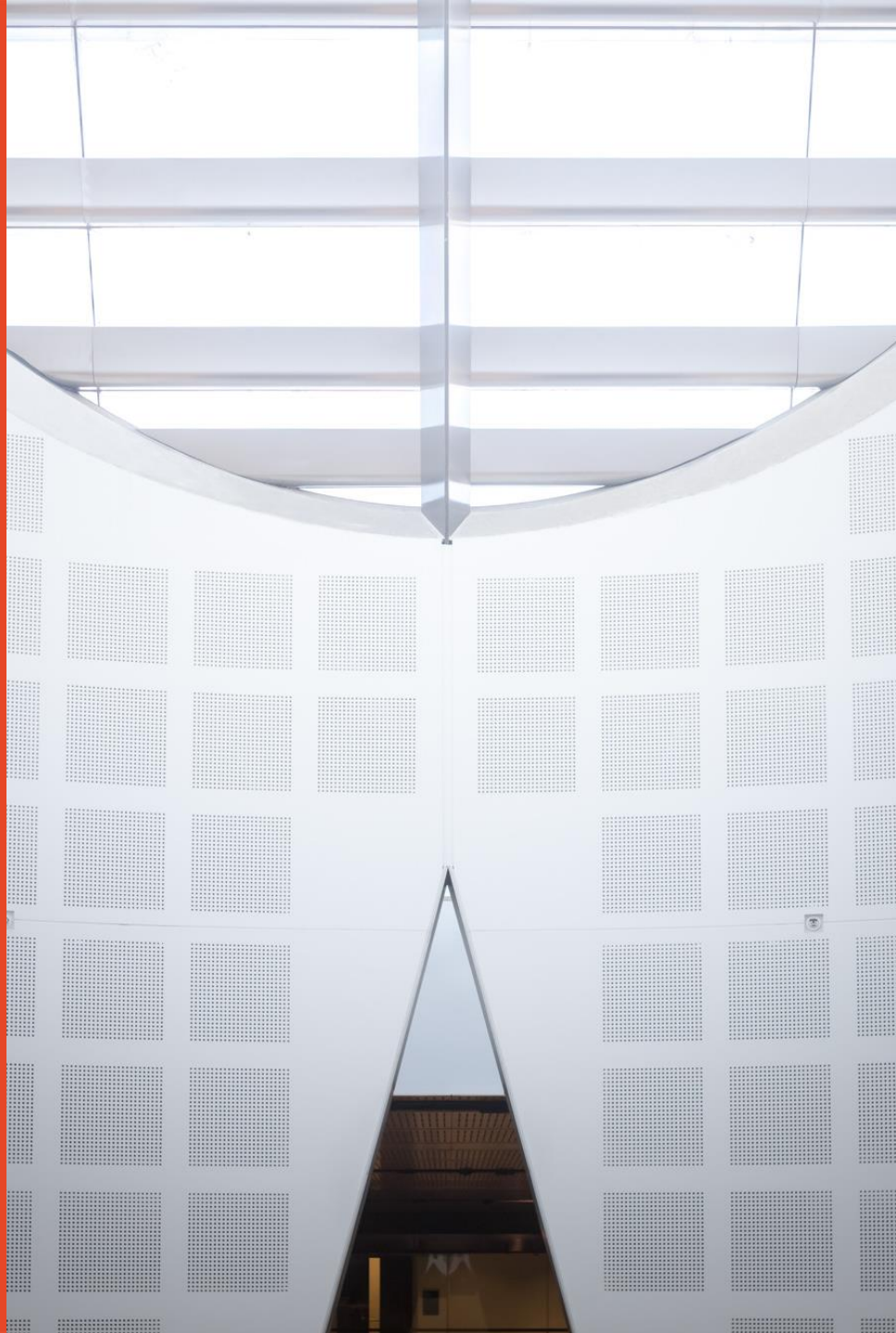
INFO1111: Computing 1A Professionalism

Week 5:
- Finding Information
- Collaboration / Git

Week 4 recap

Week 5 overview

THE UNIVERSITY OF SYDNEY



Week 4 Recap

- Tech Stacks
  - Why do different companies use different tech stacks?

- Make Files
  - Have you been able to create and execute a simple make file?
  - Test your capability:
    - Spend 5 minutes creating a make file that lets you do the following:
      - $ make setup
      - To create (only if they don't exist) the initial files for a LaTexX project (i.e. the .tex and .bib) files.

Week 4 self-learning exercise…

- Week 4 Concept: Greedy Algorithms
  - Essentially where you solve a problem by iteratively select the biggest (or fastest or …) element possible…

```
# A simple example of a greedy algorithm that determines
# the maximum weight you can put into a packing box

# weight of possible items to put in packing box
items = [20,10,5,2,1]
boxCapacity = 66
print ("Box Capacity = ", boxCapacity)

# And now the algorithm
remainingCapacity = boxCapacity
while remainingCapacity >= items[-1]: # Keep adding until the remaining capacity is
    for item in items: # search from the biggest item until we find one that will f
        if item <= remainingCapacity:
            remainingCapacity = remainingCapacity - item
            print ("Add item with weight",item,"results in remaining capacity =",re
            break
print ("No more items can be added")
```

Week 4 self-learning exercise…

- Week 4 Concept: Greedy Algorithms
  - Essentially where you solve a problem by iteratively select the biggest (or fastest or …) element possible…

```
# A simple example of a greedy algorithm that determines
# the maximum weight you can put into a packing box

# weight of possible items to put in packing box
items = [20,10,5,2]
boxCapacity = 66
print ("Box Capacity = ", boxCapacity)

# And now the algorithm
remainingCapacity = boxCapacity
while remainingCapacity >= items[-1]: # Keep adding until the remaining capacity is
    for item in items: # search from the biggest item until we find one that will f
        if item <= remainingCapacity:
            remainingCapacity = remainingCapacity - item
            print ("Add item with weight",item,"results in remaining capacity =",re
            break
print ("No more items can be added")
```

Week 5 self-learning exercise…

- Week 5 Concept: Race Condition
  - What is a race condition?
  - Under what circumstances might it be important?

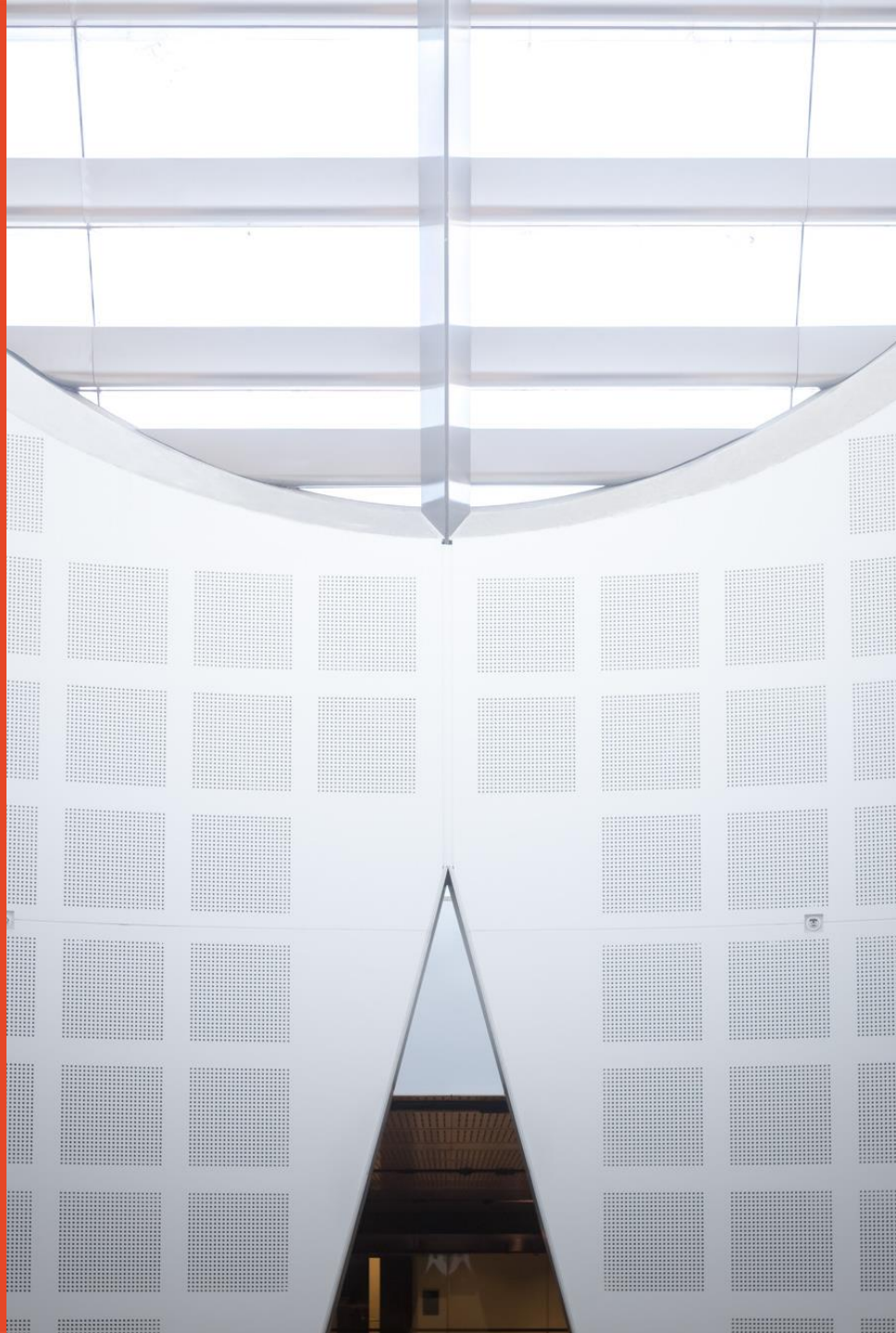
Week 5 Overview

- Finding Information
  - How good is your google fu?
  - (See how long it take you to find out my middle name?)
- Collaboration / Version Control
  - How do manage software when there are multiple versions, with numerous people working on it?
- Git + Github
  - A key tool for all software developers!

INFO1111: Computing 1A Professionalism

Week 5:
- Finding Information
- Collaboration / Git

Finding Information



THE UNIVERSITY OF SYDNEY

Finding information

- Sources
  - The materials from which ideas and information are gathered.

- Resources
  - The Web!
  - Library
    - https://library.sydney.edu.au/
    - https://library.sydney.edu.au/research/index.html?accordion=findinfo
    - https://library.sydney.edu.au/databases/
  - Google Scholar
    - https://scholar.google.com.au/
  - ACM Digital Library
  - https://dl.acm.org/
  - Many other sources of information accessible online

Evaluating Sources

- **"Should a data scientist learn git?**
- Absolutely. At minimum a data scientist should know at least one repository tool and git is perhaps the most leading tool. Without knowing git, it is very hard to share your work with other team members. Also git repository is likely to be automatically backed up so you are relieved of that responsibility."

  - https://www.quora.com/Should-a-data-scientist-learn-git

Evaluating Sources

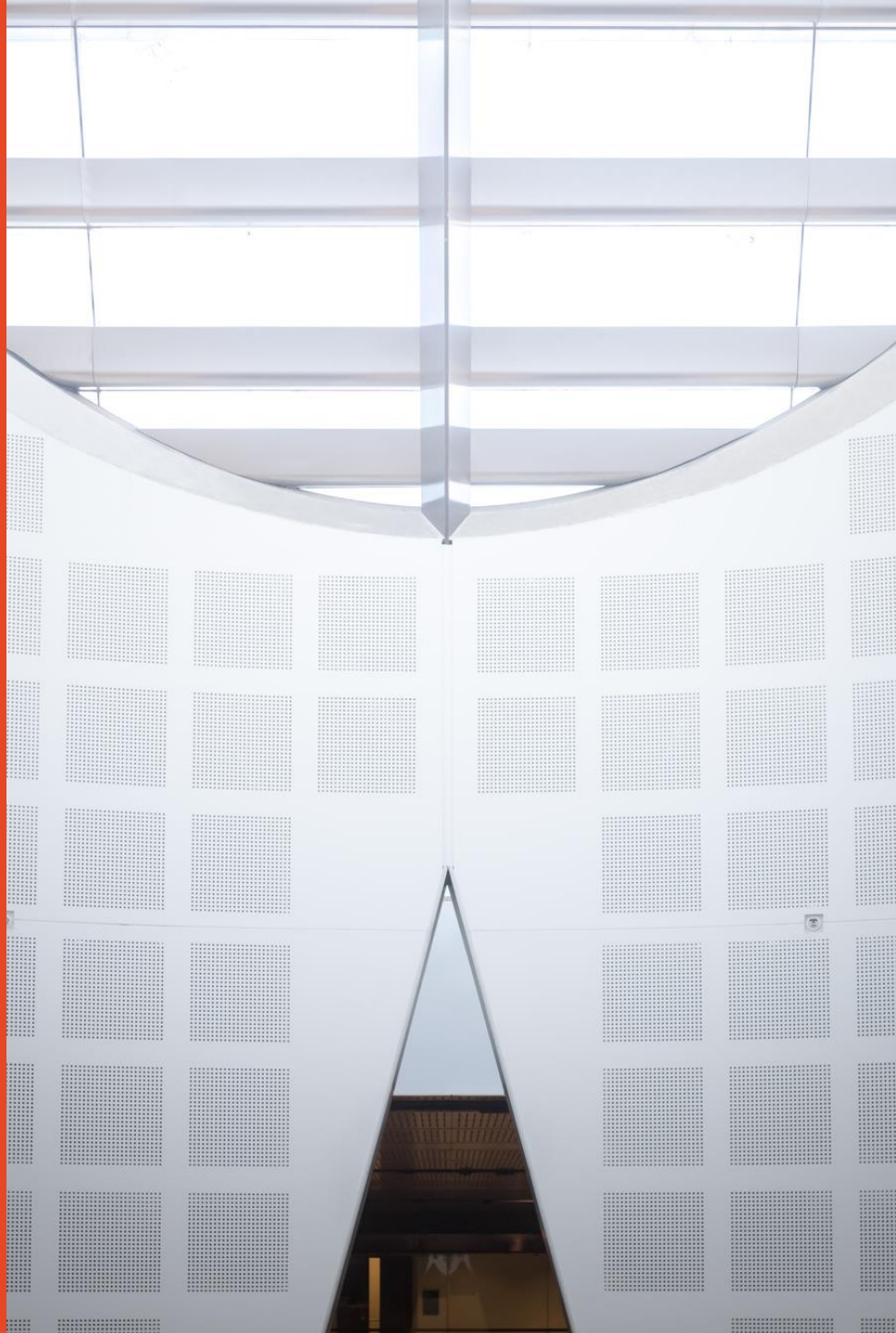
	'REVIEW' CRITERIA	'REVIEW' QUESTIONS
R	**Relevance**	Is the information in the source directly relevant to the requirements for your report?
E	**Expertise of Author**	1. What are their qualifications? 2. Are they writing in their area of expertise? 3. Are they cited by other authors in the field?
V	**Viewpoint of Author/Organisation**	1. Does the author have any personal or professional affiliations that may bias their work (e.g. a blog written by an IT supplier)? 2. Has the information been sponsored by an organisation who might gain from publishing the information (as above)? 3. What is the purpose of the source – to inform, persuade, or entertain?
I	**Intended Audience**	Is the source aimed at the general public, professionals, or scholarly audience?
E	**Evidence**	1. Are statements supported by evidence? - Primary sources (*My data shows…)* - Secondary sources (*David claimed that his data showed …)* - Expert opinion (*Prof Smith thinks that …)* - Uninformed opinion (*Someone on the bus told me …*, I read a blog that said) 2. Are the references to the evidence correct? 3. Has the source been subjected to peer review?
W	**When Published**	1. When was the source published? 2. Have significant developments been made in the subject area since the source was published?

INFO1111: Computing 1A Professionalism

Week 5:

- Finding Information
- Collaboration / Git

Referencing



THE UNIVERSITY OF SYDNEY

Finding information: Referencing

- "Referencing is an essential part of academic writing. Its purpose is to acknowledge the original source of ideas and work that is not the author's own and to point the reader to the original documents so that they can determine independently whether the attributed sources support the author's argument as written." - http://libguides.library.usyd.edu.au/citation

- This is an example of a direct quotation.
- You can also *paraphrase* an author or use ideas from their earlier works both of which must be cited properly at the end of your document.
- Your own previous work must be cited properly as well.
- Proper referencing is part of academic integrity.

Finding information: Academic integrity

- You can research, use, analyse, critique and compare the work of others as long as you reference it properly.

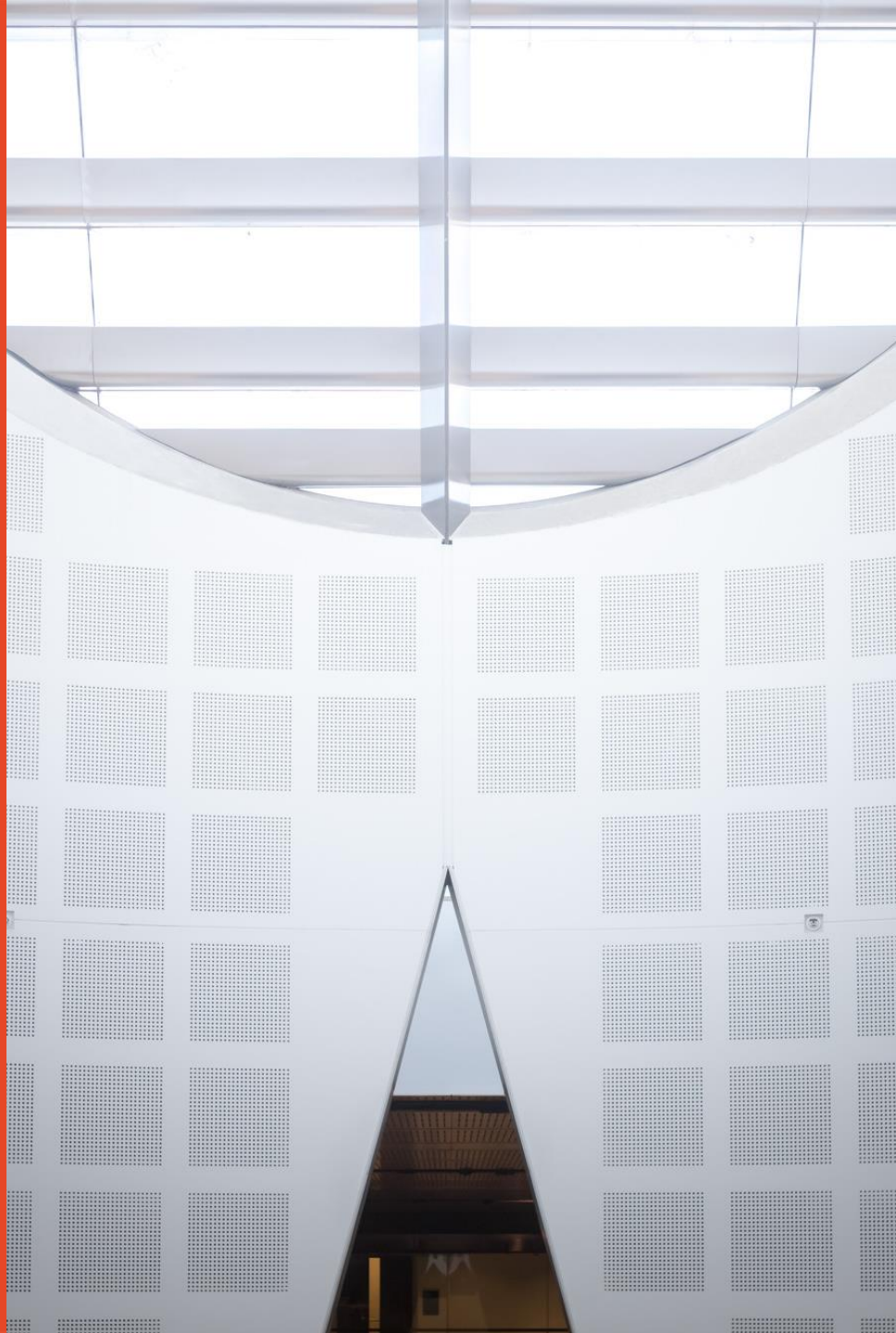
- BUT
- You cannot copy other people's work either word-for-word or by paraphrasing without acknowledgement
- You cannot recycle work you had previously published
- You cannot fabricate data (e.g. fake results in a lab report)
- You cannot submit the work of others as your own
- You cannot knowingly assist others in an act of academic dishonesty

- You must make it clear what is your work (and ideas!) and what has come from others…

INFO1111: Computing 1A Professionalism

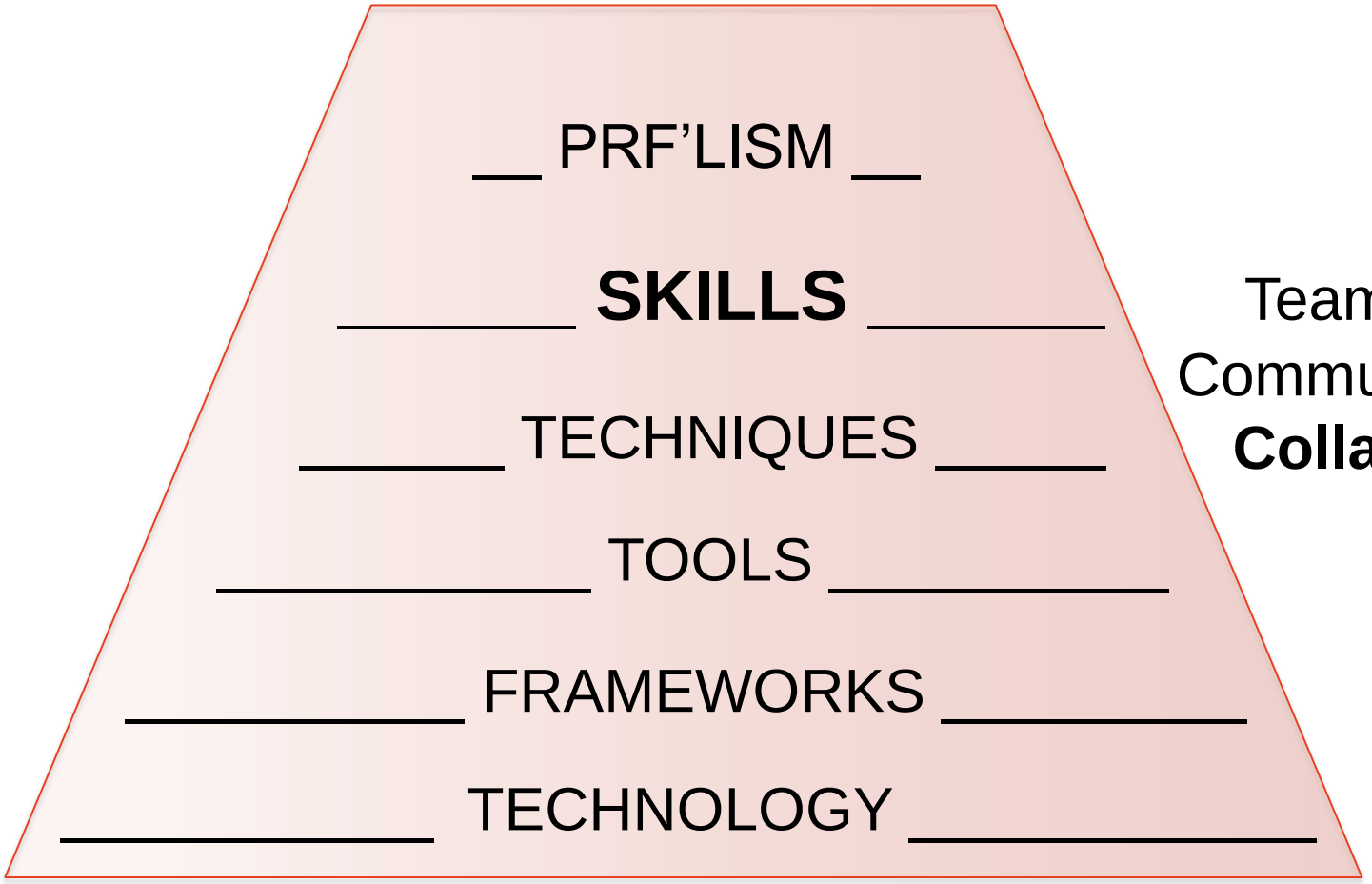
Week 5:
- Finding Information
- Collaboration / Git

Collaboration



THE UNIVERSITY OF SYDNEY

Skills – Team work



Teamwork,
Communication,
Collaboration

Collaboration

- What is collaboration?
- "to work jointly with others or together especially in an intellectual endeavor"

  ~ "Collaborate." *Merriam-Webster.com*. Merriam-Webster, n.d. Web. 27 Mar. 2018.
- "the situation of two or more people working together to create or achieve the same thing"

  ~ Definition of "collaboration" from the Cambridge Advanced Learner's Dictionary & Thesaurus © Cambridge University Press
- "the process of two or more people or organizations working together to complete a task or achieve a goal..."

  ~ Wikipedia
- ... etc

Collaboration

- Collaboration for students
  - Group work
  - Attribution
  - Maximising output with limited resources
  - *Look these up and complete*

- Collaboration for IT professionals
  - Not so different …
  - Higher stakes, costlier mistakes
  - Traceability
  - Attribution

Collaboration – IT Industry Requirements

- Increasingly, systems and processes are being set up for sharing and collaboration …
    - Online tools (e.g. Asana)
    - Shared repositories
    - Discussion forums
- Professional practice has additional requirements:
    - Security
    - Traceability
    - Accountability

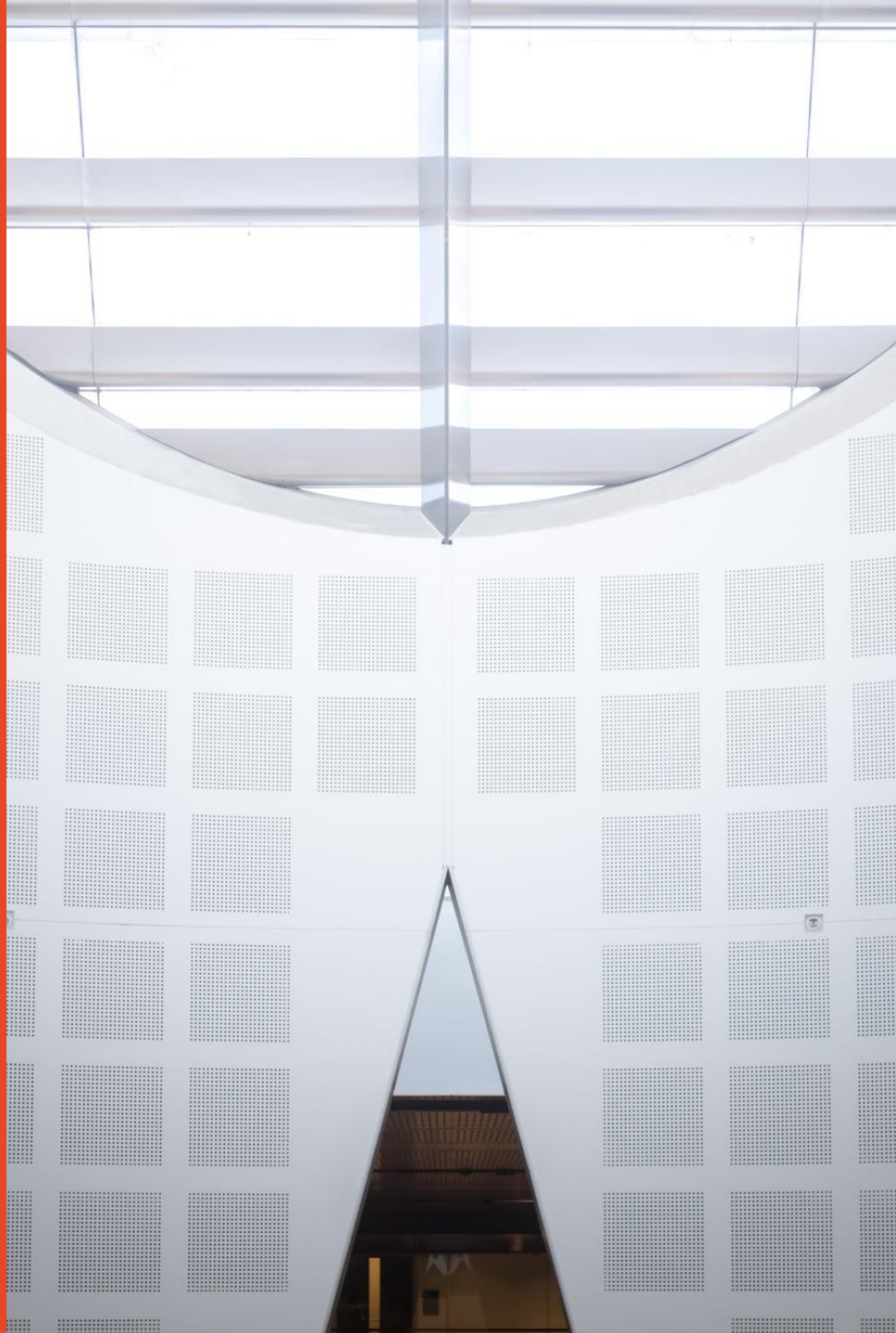
    - AND version control

INFO1111: Computing 1A Professionalism

Week 5:

- Finding Information
- Collaboration / Git

Version Control

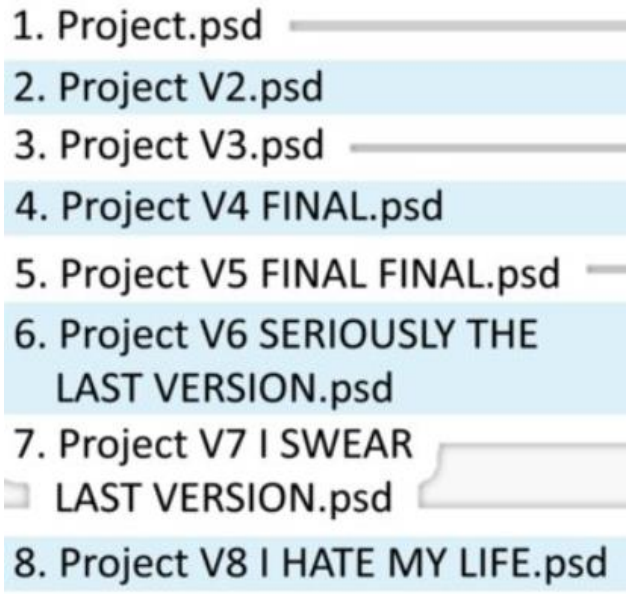


THE UNIVERSITY OF SYDNEY

How do multiple people work on the same artefact?

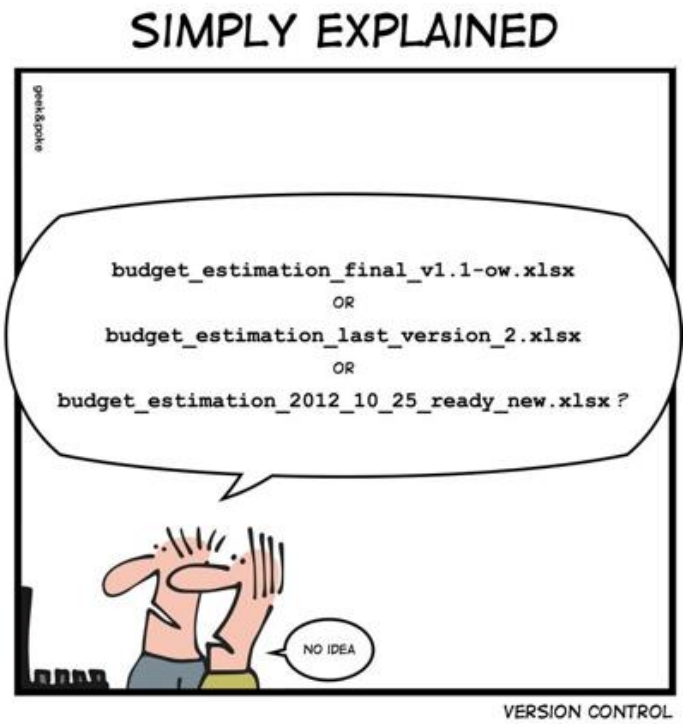
- Example:
  - Multiple people can edit a GoogleDoc file
  - Multiple people can work on the same codebase

- How does that work?
- Who wrote what?
- Problems
  - Traceability
  - Version history
  - Security
  - …



How do multiple people work on the same artefact?

- Using different filenames can be a real problem

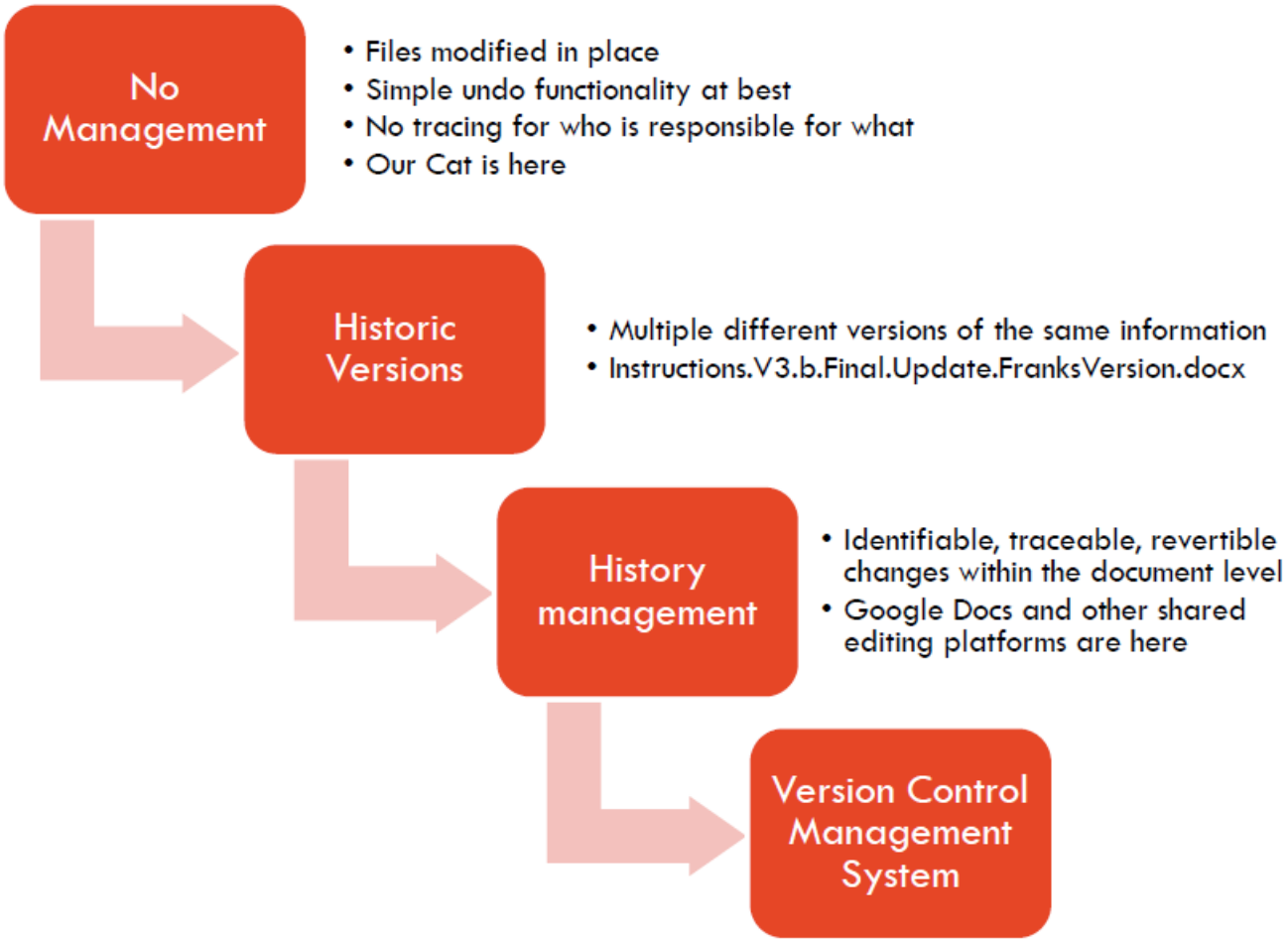


https://www.datamation.com/news/tech-comics-version-control-1.html

How do multiple people work on the same artefact?

- Why is version control important?
- Versioning
  - A bug was added in – but how do I return to the point where that occurred?
    - *Who here thinks they could edit 100 lines of code, and then undo those edits without using ctrl-Z?*
  - And what if there are cross-file dependencies?
- Clashes
  - Parallel edits to different versions of the same document set?

An Incomplete History Of Information Management Systems



Version control systems

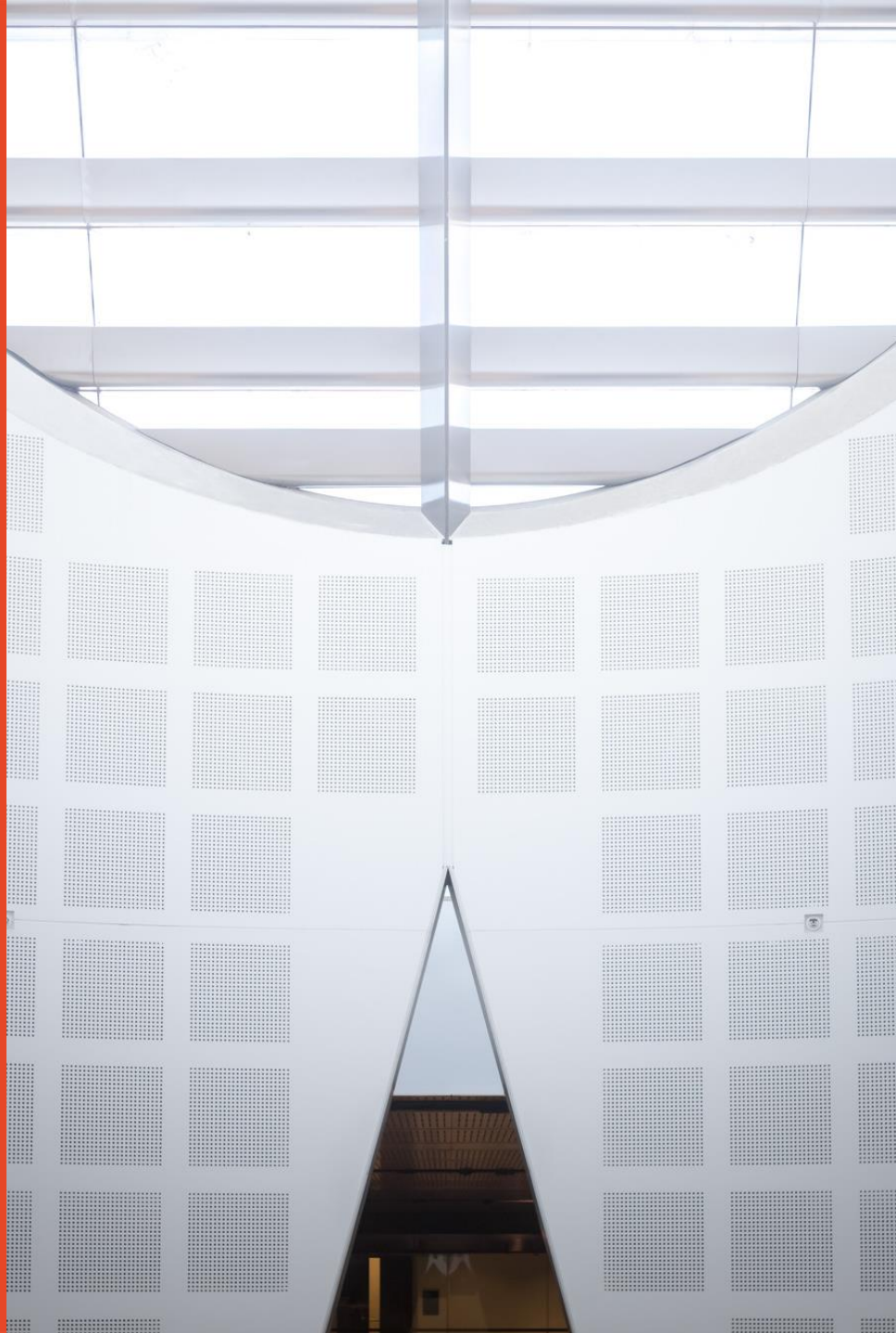
- Concepts (software) date back to the 1970's
  - RCS – 1982
  - CVS – 1986
  - SVN / Subversion – 2004
  - Git – 2005
  - Mercurial – 2005
  - See
    - https://trends.google.com/trends/explore?date=all&q=git,svn
    - https://rhodecode.com/insights/version-control-systems-2016
- Various front-ends to make their use easier…
  - Bitbucket
  - Github

INFO1111: Computing 1A Professionalism

Week 5:

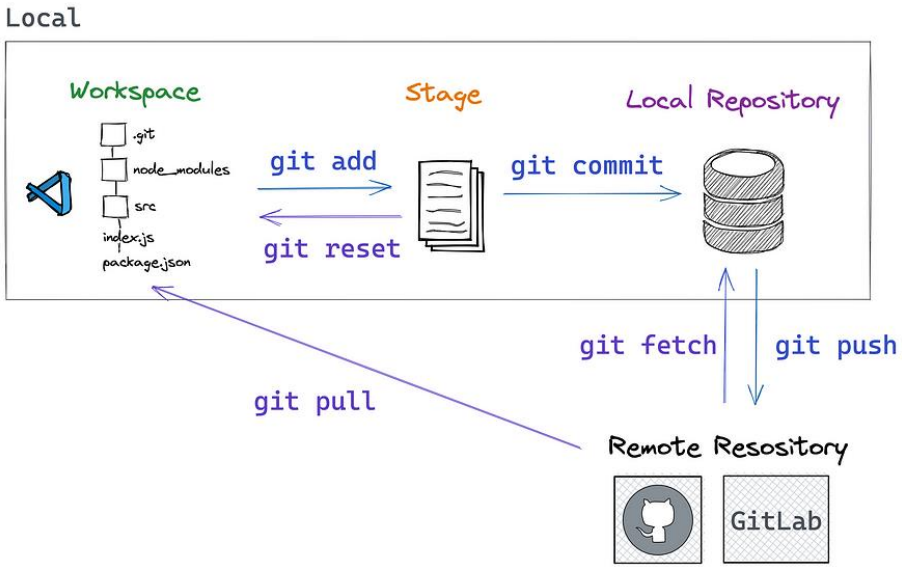
- Finding Information
- Collaboration / Git

Git

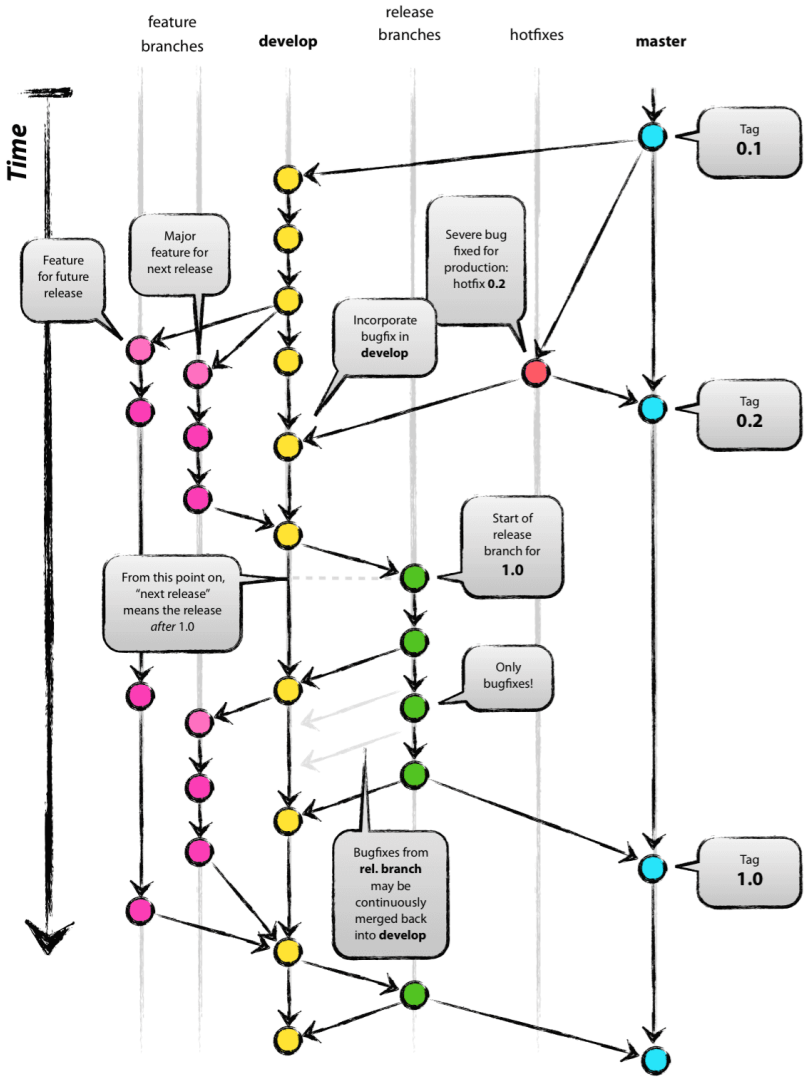


THE UNIVERSITY OF SYDNEY

Collaboration – Git

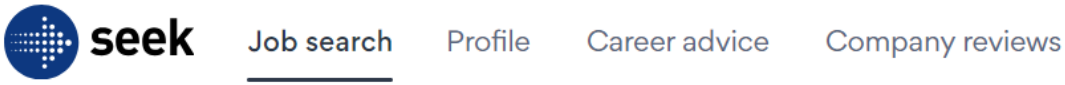


From https://medium.com/frontend-canteen/you-can-master-git-git-commands-with-these-diagrams-40a0b2f5cc42



From: https://nvie.com/posts/a-successful-git-branching-model/

Collaboration – Git



Junior Software Developer

Our main goal then is to hire people who are switched on and can work autonomously but are also able to engage with members of the team, especially in other disciplines.

While none of these are mandatory we would consider bonus points for any of the below:

- Portfolio demonstrating game development chops or software engineering wizardry! (EX++ score for awesome and diverse portfolios)
- Experience with Unity, Unreal, or other Game Engines;
- Working knowledge of, or experience working with exports from, 2D or 3D animation packages, such as Spine, Maya, Blender, 3dsMax;
- Experience with issue and source control tools like JIRA and Git.
- Degree in Computer Science or equivalent;
- Degree or Diploma majoring in Games Development or equivalent;

Fresh graduates, please provide your academic transcript in your application.

Please provide a short cover letter telling us why you're the right person for our team and which one of our games you liked.

What is Git?

- Git works using a database of snapshots
- These snapshots are effectively what your information (directories, files, and contents) looked like at various points in time
- These points in time are called ‘commits’
- To create a commit, you modify files (i.e., do some work), ‘stage’ them ready to be committed, and then commit your staged work
- You can rewind and fast forward time back and forth between these commits
- You can ‘branch’ from the main flow of this history and ‘merge’ back in
- You can ‘push’ your work from your computer to a remote storage location and ‘pull’ other people’s work from a remote storage location to your computer

Remote Storage Location

- Git is primarily intended for multiple people to collaborate
  - This requires some way of getting code from Machine A to Machine B
- Git doesn't require a central location
  - (remember it is a distributed version control system, not a centralized one), but in practice it will often use one
- You can host your own git server
- Several popular remote storage server providers also exist
  - GitHub
  - BitBucket
  - GitLab
  - AWS Commit
  - Azure DevOps
- In this unit we will be using https://github.sydney.edu.au

Collaboration – Git

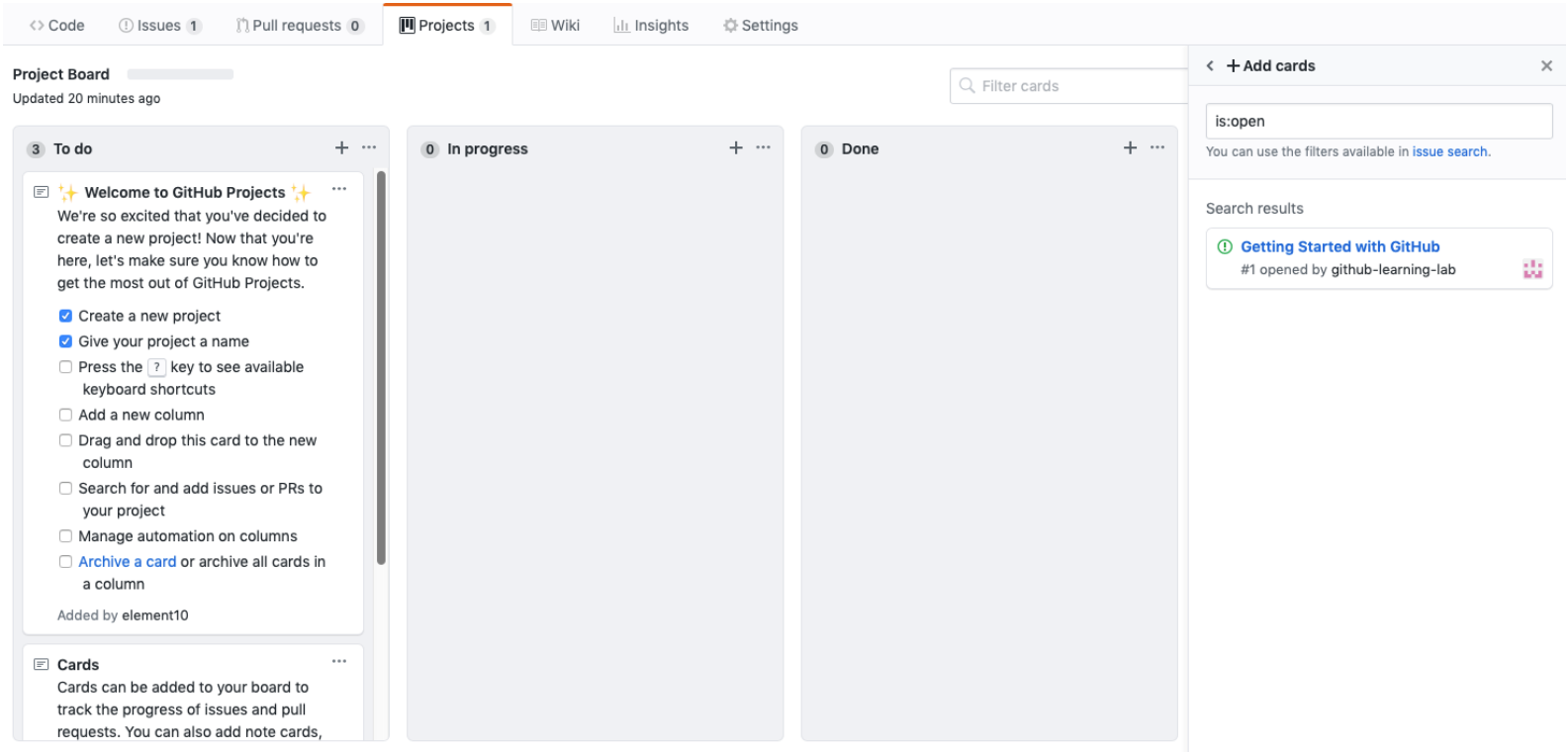
- View of git – commands and work flow
  - See https://try.github.io/levels/1/challenges/1
  - See https://www.learnenough.com/git-tutorial

Collaboration – Project Management

- Important: task allocation, transparency, efficiency
- To-Do lists
- Multiple methodologies
  - Lean
  - Agile
  - Scrum
  - Waterfall
  - Kanban
  - etc…

Collaboration – Project Management

- Kanban boards on Github

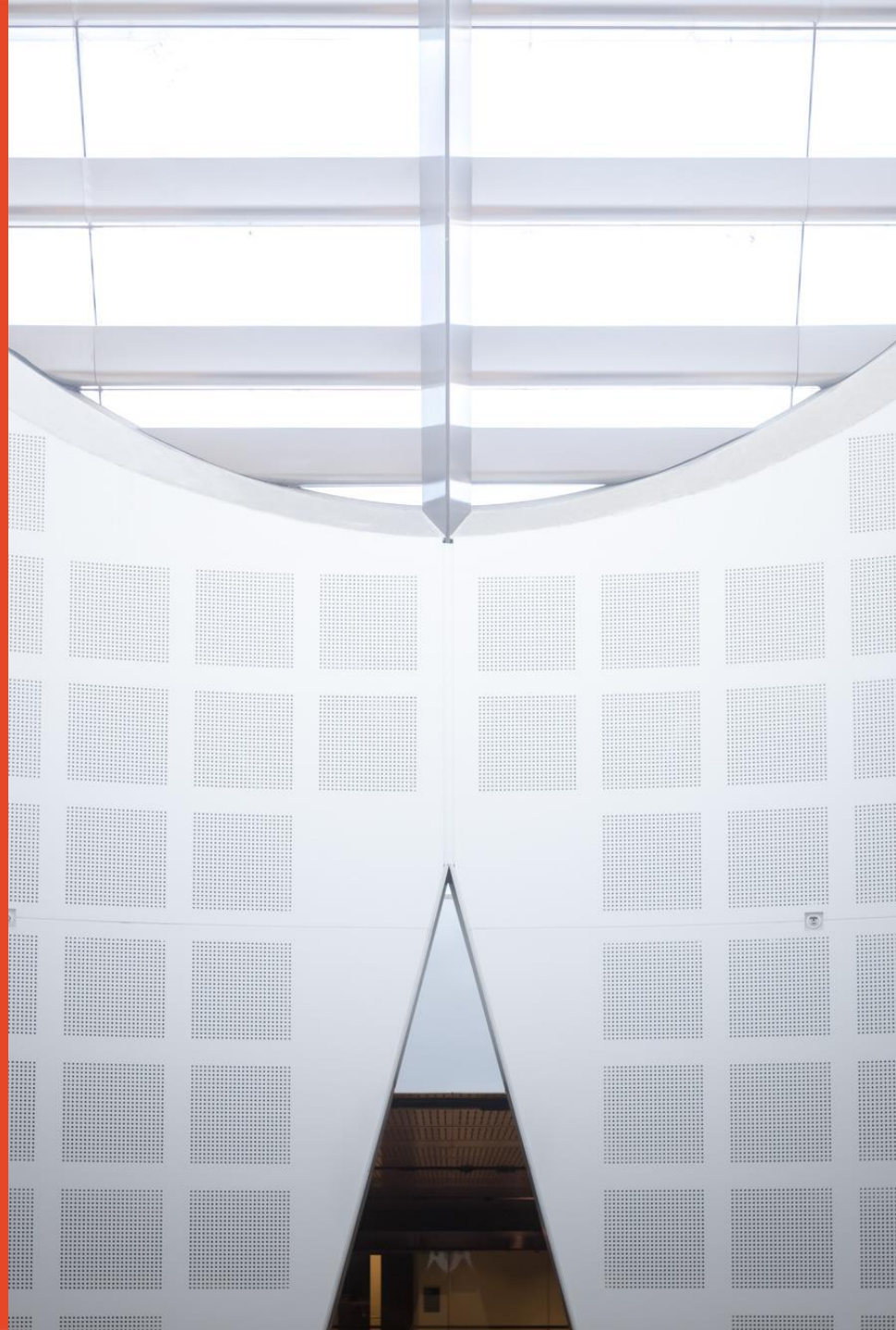


INFO1111: Computing 1A Professionalism

Week 5:

- Finding Information
- Collaboration / Git

Worked Example

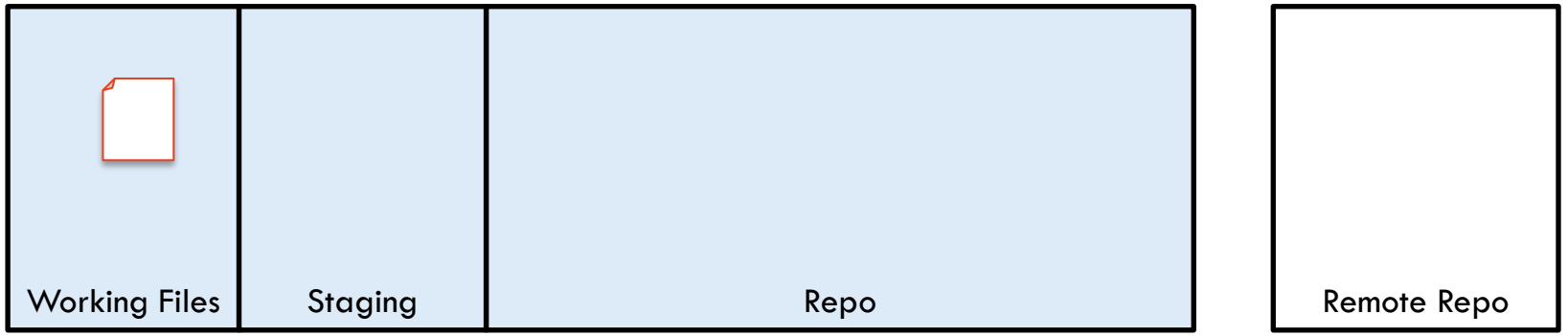


THE UNIVERSITY OF SYDNEY

Git - Worked example… 1

1. Create a simple HTML page, initialise the repo … and commit into the repository

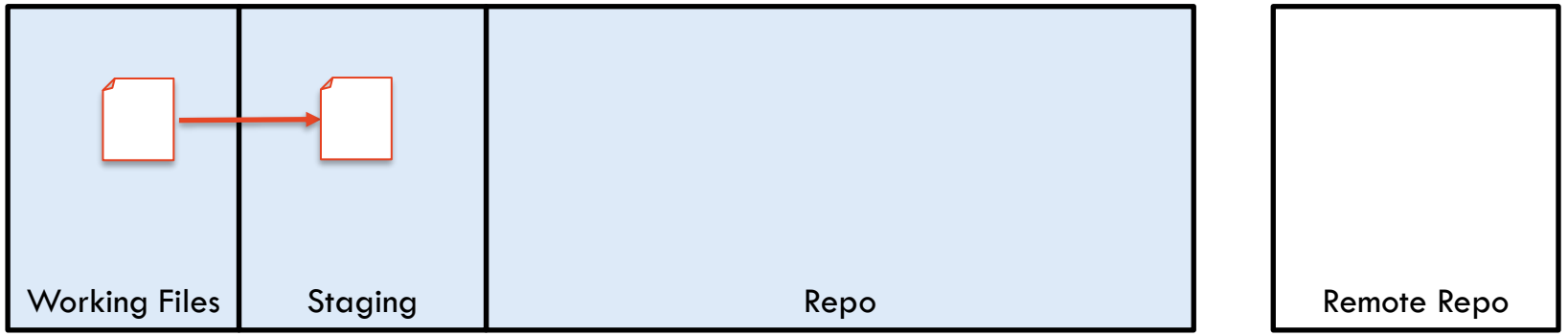
```
> edit paper.html
> git init
```



Git - Worked example… 1

1. Create a simple HTML page, initialise the repo … and commit into the repository

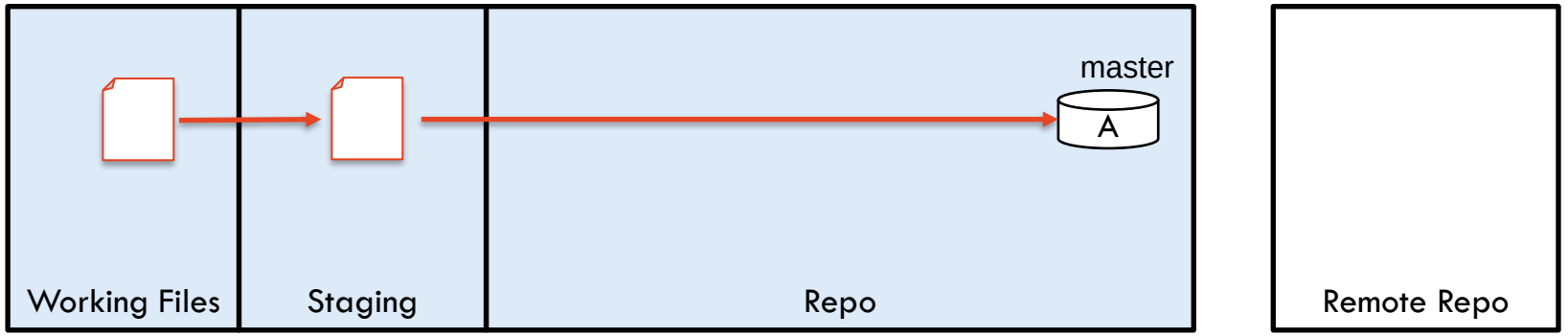
```
> edit paper.html
> git init
> git add paper.html
```



Git - Worked example… 1

1. Create a simple HTML page, initialise the repo … and commit into the repository

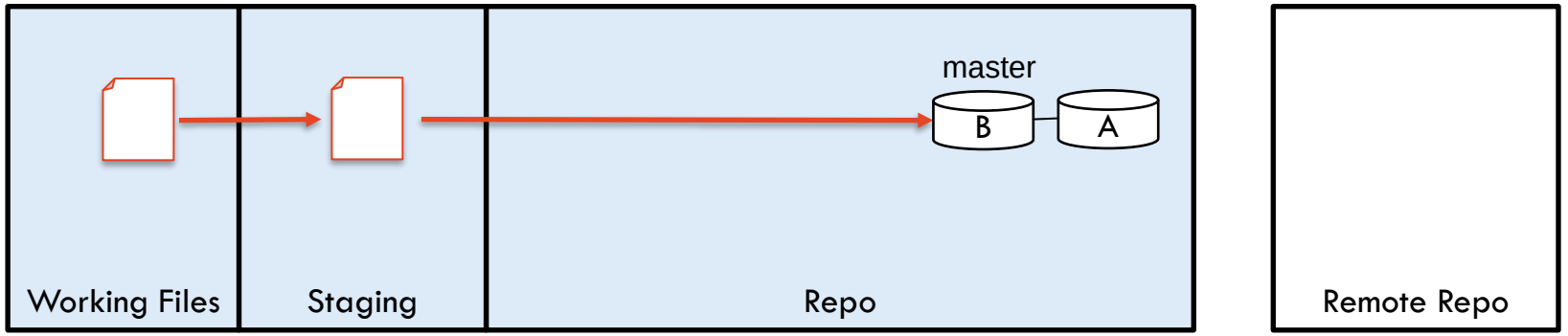
```
> edit paper.html
> git init
> git add paper.html
> git commit –m “First paper”
> git status
> git log
```



Git - Worked example… 2

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit

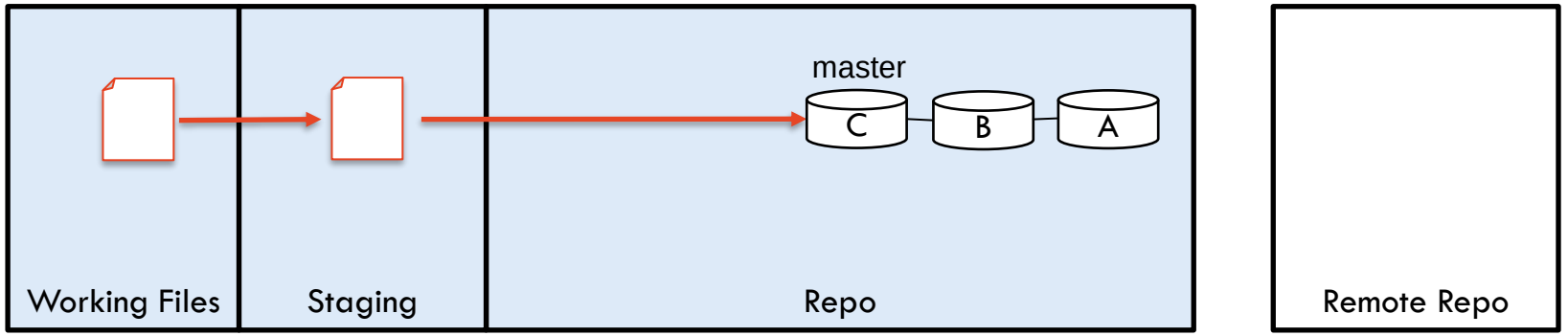
```
> edit paper.html  (add some changes)
> git add paper.html
> git commit –m “Added styling”
```



Git - Worked example… 3

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit

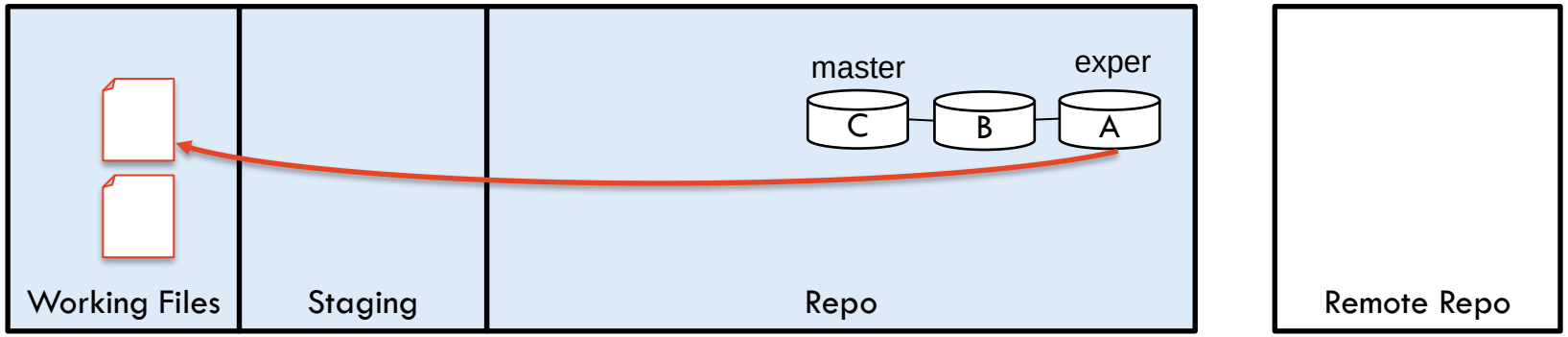
```
> edit paper.html  (add some changes)
> git diff
> git add paper.html
> git commit –m “Multicolour styling”
```



Git - Worked example… 4

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit

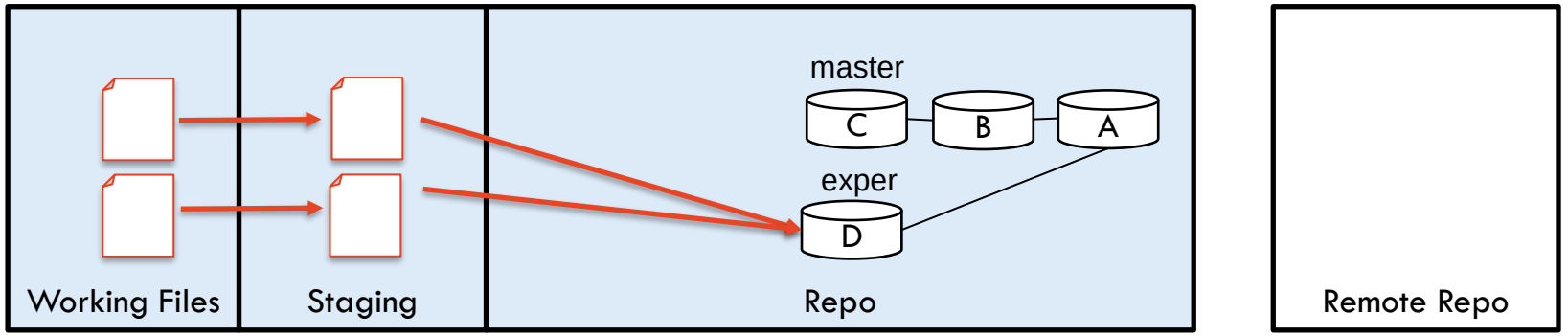
```
> git checkout –b exper <commitA>
> edit paper.html
> edit paper.css     (create a new file)
```



Git - Worked example… 4

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit

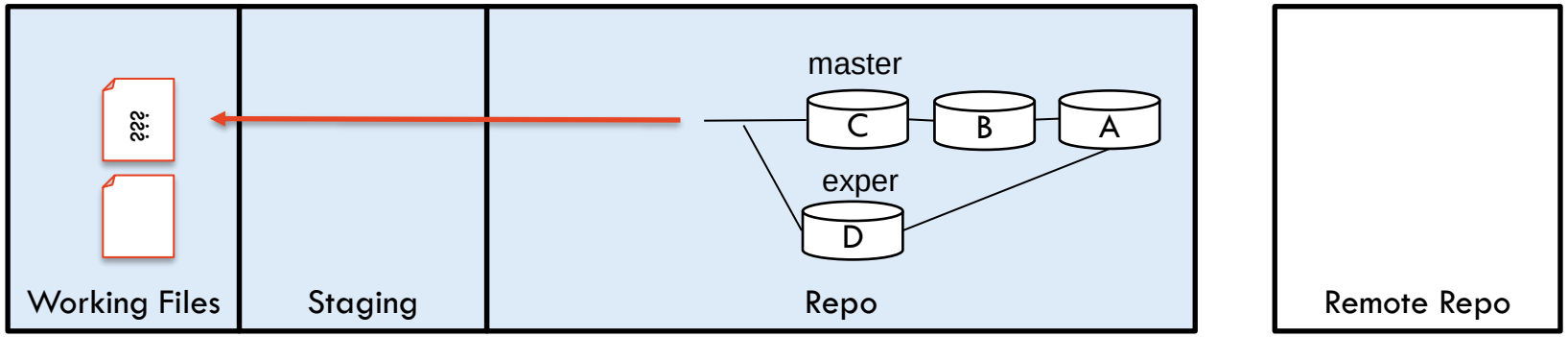
```
> git checkout –b exper <commitB>
> edit paper.html
> edit paper.css

> git add paper.*
> git status
> git commit –m “Seperate CSS”
> git log
```



Git - Worked example… 5

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit
5. Merge the branch back in.

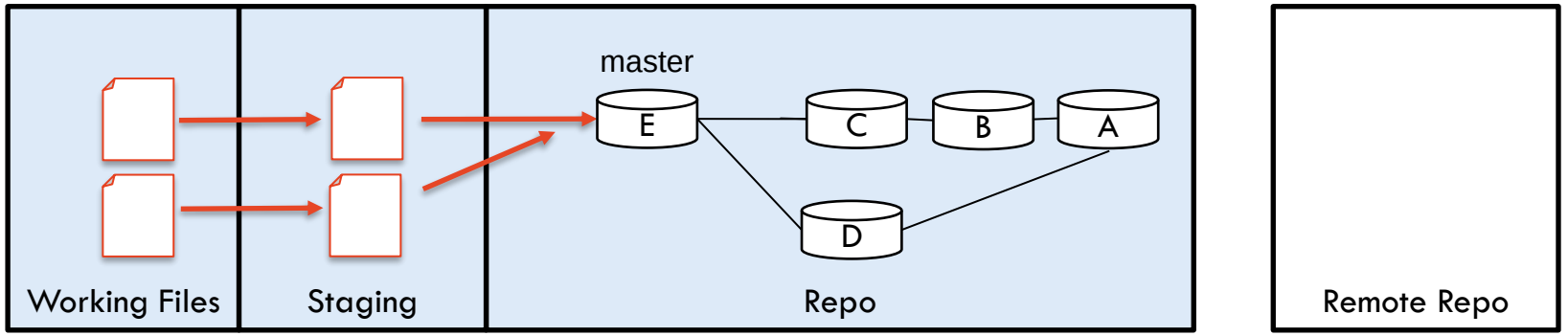
```
> git branch
> git checkout master
> git merge exper  (will flag conflicts)
> git status
```



Git - Worked example… 5

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit
5. Merge the branch back in.

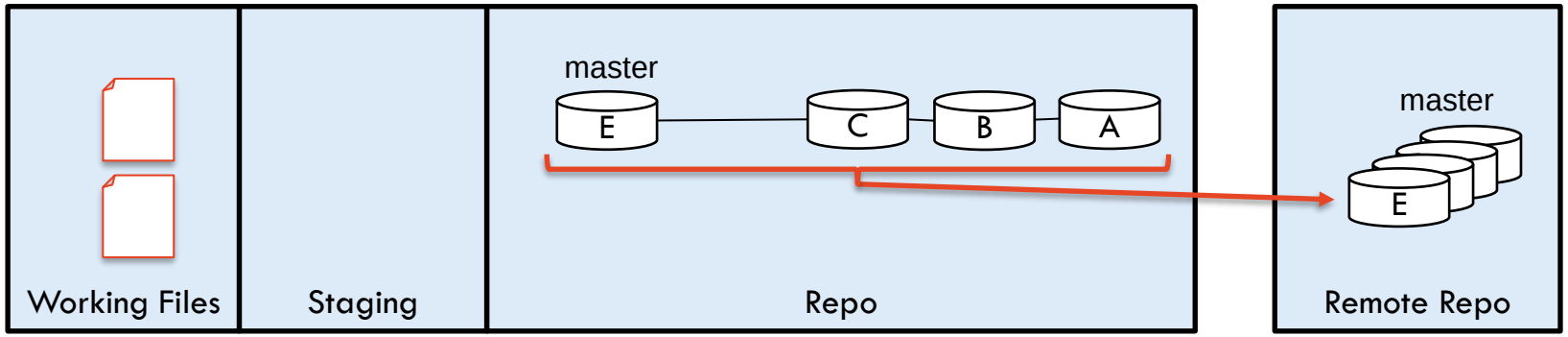
```
> git branch
> git checkout master
> git merge exper   (will flag conflicts)
> git status

> edit paper.html    (fix conflicts)
> git status
> git add paper.html
> git commit –m “Merged version”

> git branch –d exper
```



Git - Worked example… 6

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit
5. Merge the branch back in.
6. Create a remote repository and push

```
> git remote add origin
https://github.sydney.edu.au/INFO1111-2024/GitDemo.git

> git push –u origin --all
```

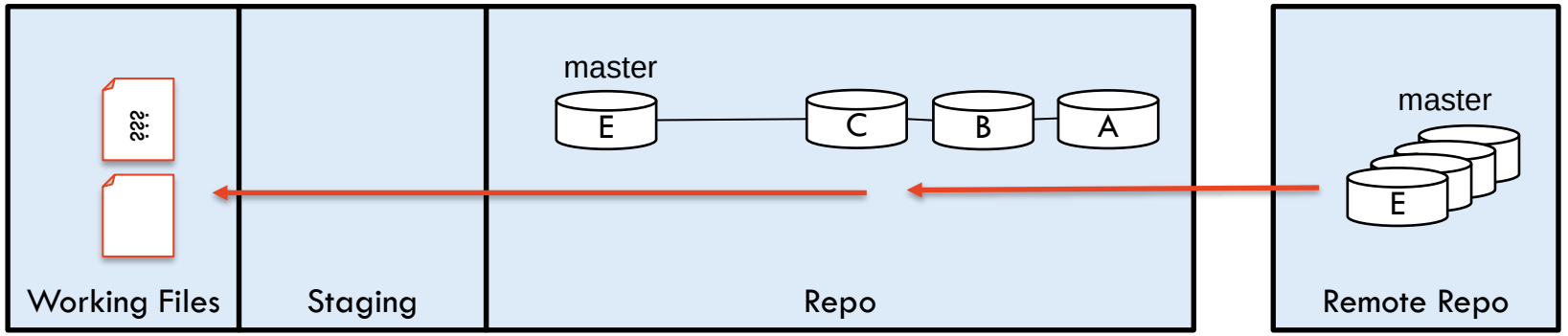


Git - Worked example… 7

1. Create a simple HTML page, initialise the repo … and commit into the repository
2. Add simple styling to the file (red text) … and commit
3. Modify so that there are different coloured paragraphs … and commit
4. Create a branch (from step 2) to try a different approach to use CSS … and commit
5. Merge the branch back in.
6. Create a remote repository and push
7. Then multiple people make changes….

```
> git clone
https://github.sydney.edu.au/INFO1111-2022/Lecture3.git

… edit / add / commit
> git push –u origin
```

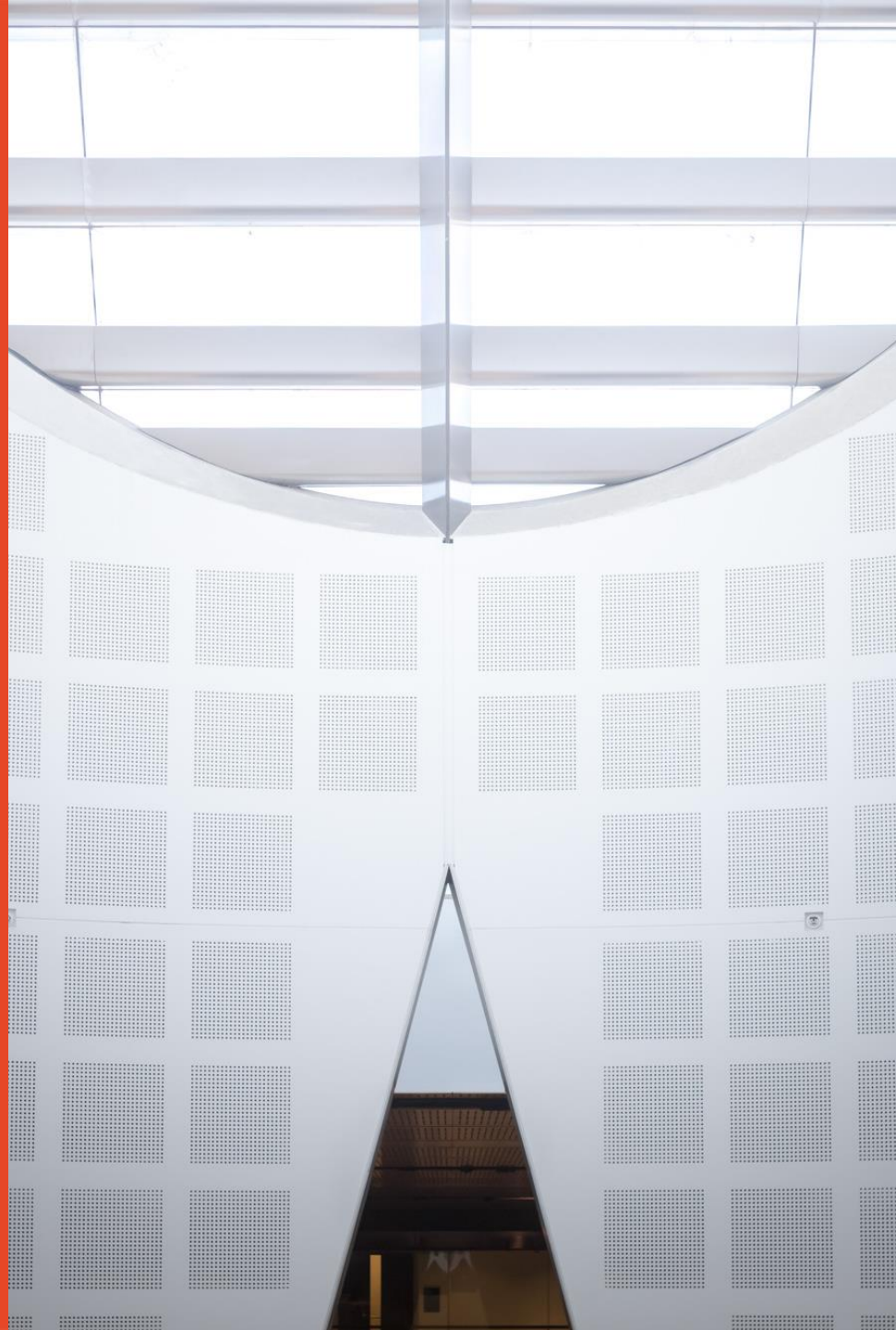
```
… edit / add / commit
> git push –u origin

aaggghhhhhhhhhhh
```



INFO1111: Computing 1A Professionalism

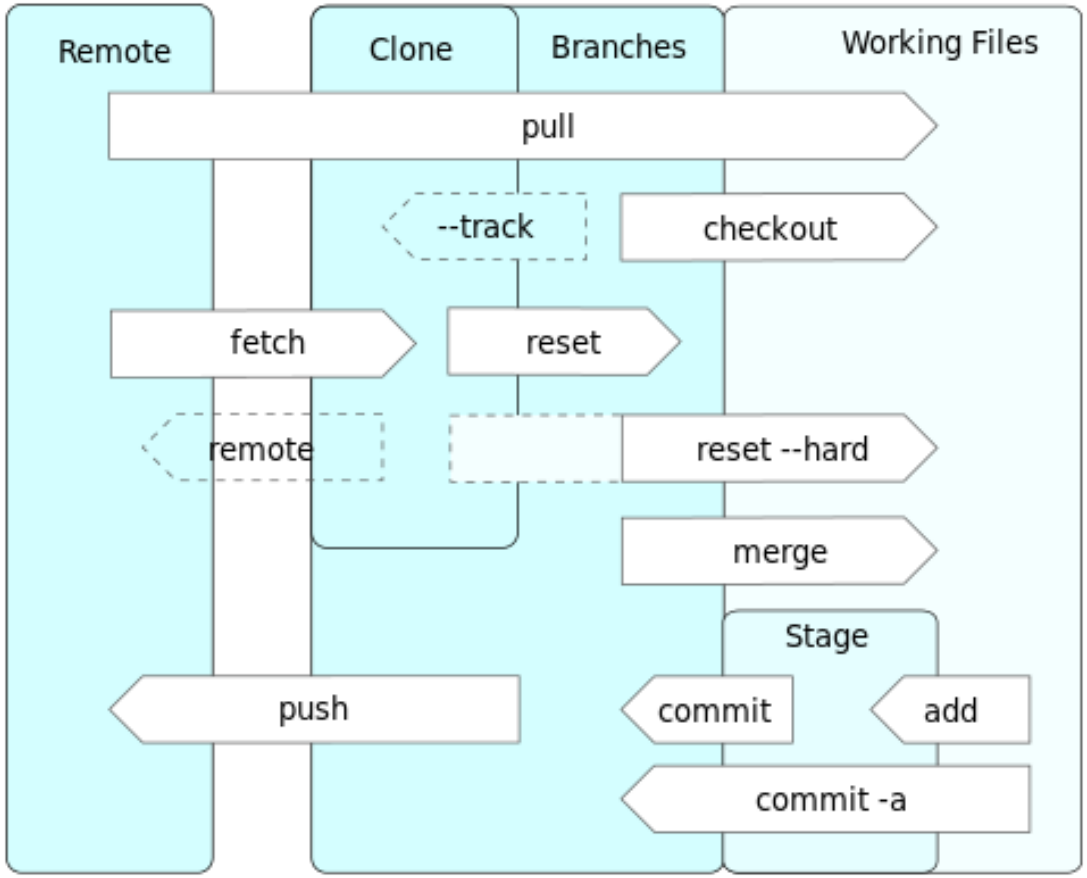
Week 5:
- Finding Information
- Collaboration / Git

Git wrap-up



THE UNIVERSITY OF SYDNEY

Collaboration – Git



Collaboration

- And then after the lecture have a look at:
  - https://betterexplained.com/articles/a-visual-guide-to-version-control/
  - https://try.github.io/levels/1/challenges/1
- And then have a read of the following, and see which side of the debate you think you might agree with…
  - https://news.ycombinator.com/item?id=5768802

- (And have a watch of https://vimeo.com/15943704)