Coding: Best Practices & Tools

Towards painless programming for researchers

L. Manzari – M. Gaborit

MWL Lunch Seminar – June 1, 2016

Contents

1. Better practices in coding: why?
2. Editors & IDEs
3. Versioning for fun and profit
4. Documentation
5. What's next?

Better practices in coding: why?

Better practices in coding: why?

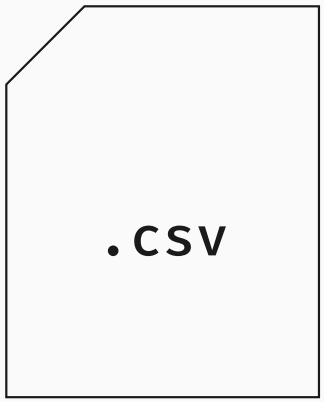
Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

— D. Knuth

Editors & IDEs

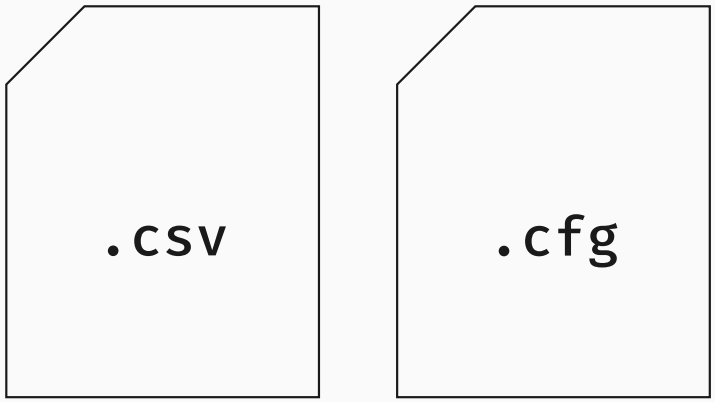
File types in research

File types in research



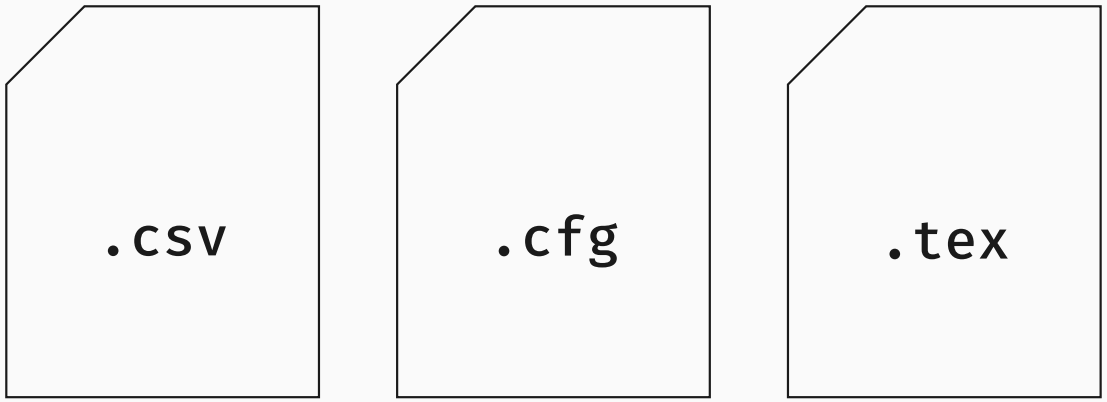
data

File types in research



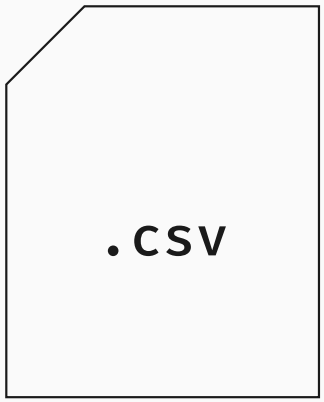
data config

File types in research

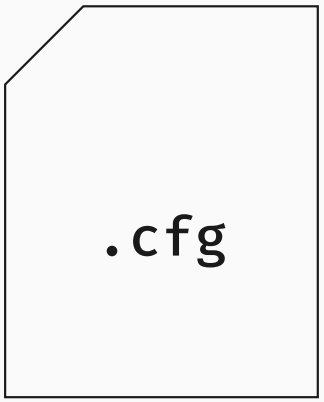


data config LaTeX

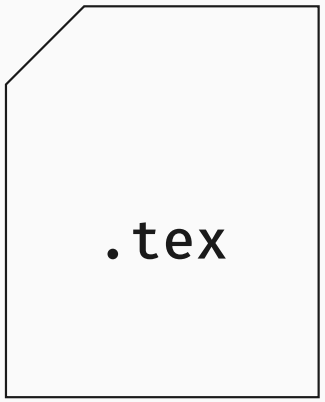
File types in research



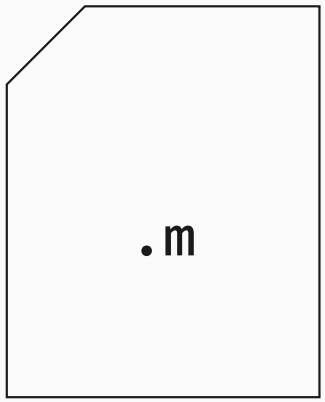
data



config

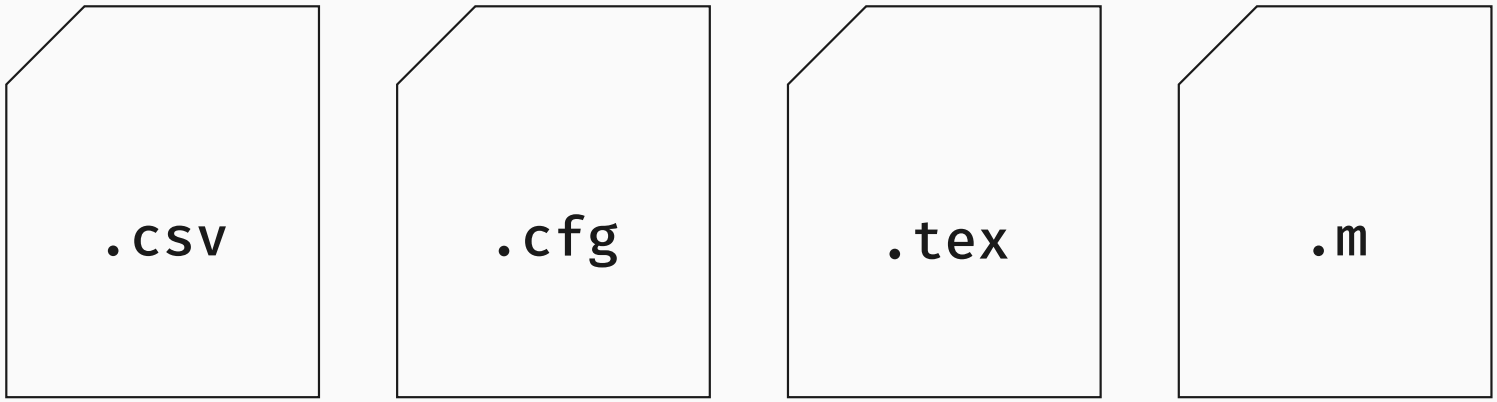


LaTeX



code

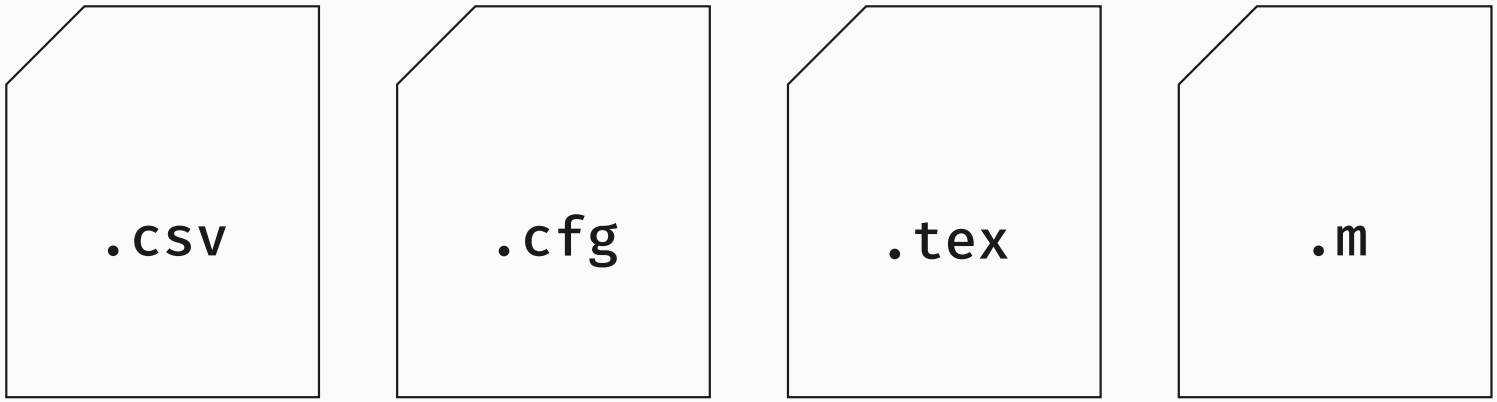
File types in research



data config LaTeX code

These are all text files...

File types in research



data config LaTeX code

These are all text files...

...let's use a good text editor!

What makes a text editor *good*?

A good text editor has to be...

- Invisible
- Intelligent
- Customizable
- Extensible

All the fuss about VIm and GNU Emacs

Pros

- learn once, use forever
- available everywhere
- (im)proved with time

Cons

- at the beginning, intimidating

Editors: GNU Emacs

Characteristics

- It’s not a text editor, it’s a Lisp interpreter
- Highly context-dependent behavior

Editors: GNU Emacs

Characteristics

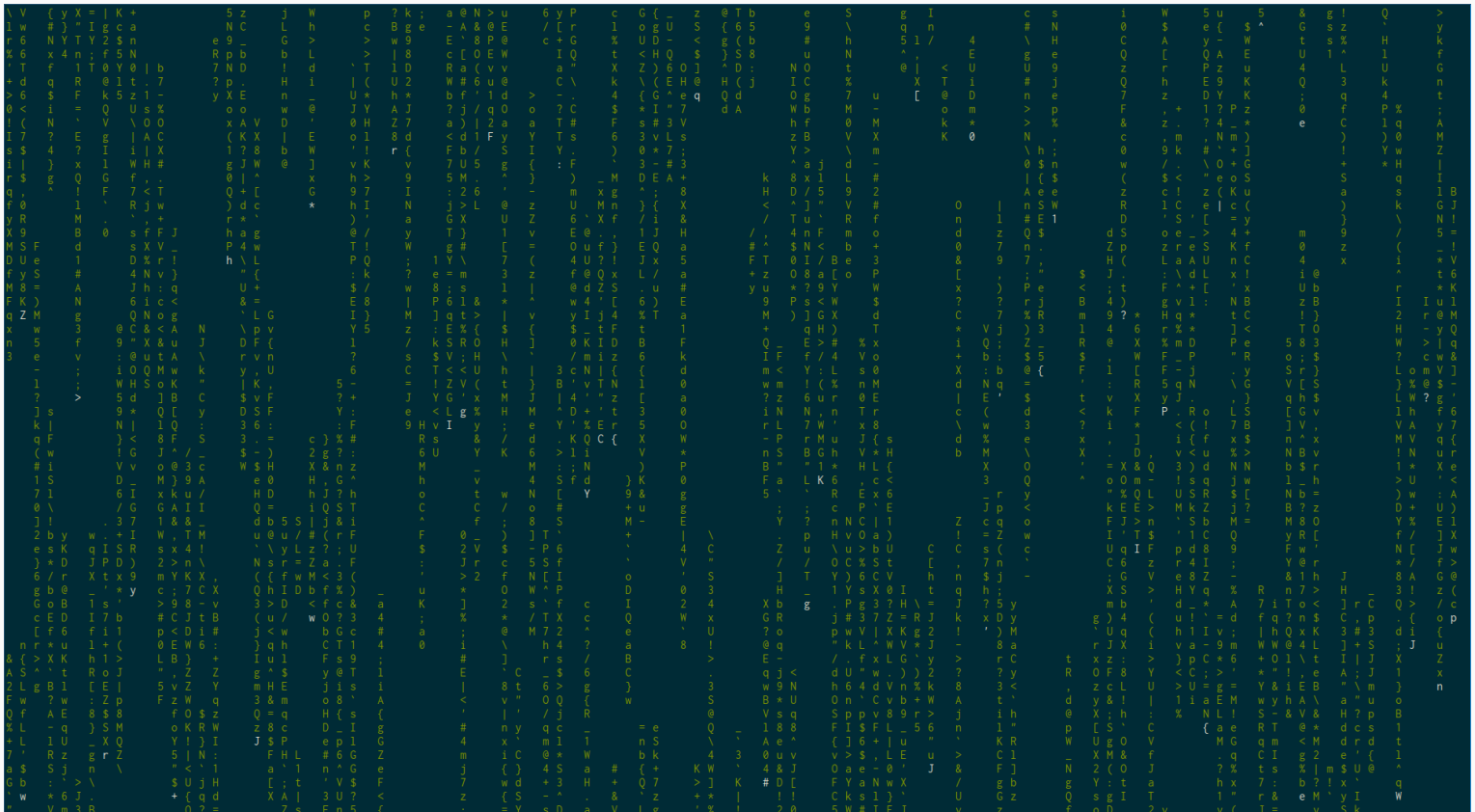
- It's not a text editor, it's a Lisp interpreter
- Highly context-dependent behavior

How to get started

1. use it as a regular text editor
2. customize according to need
3. extend it using packages

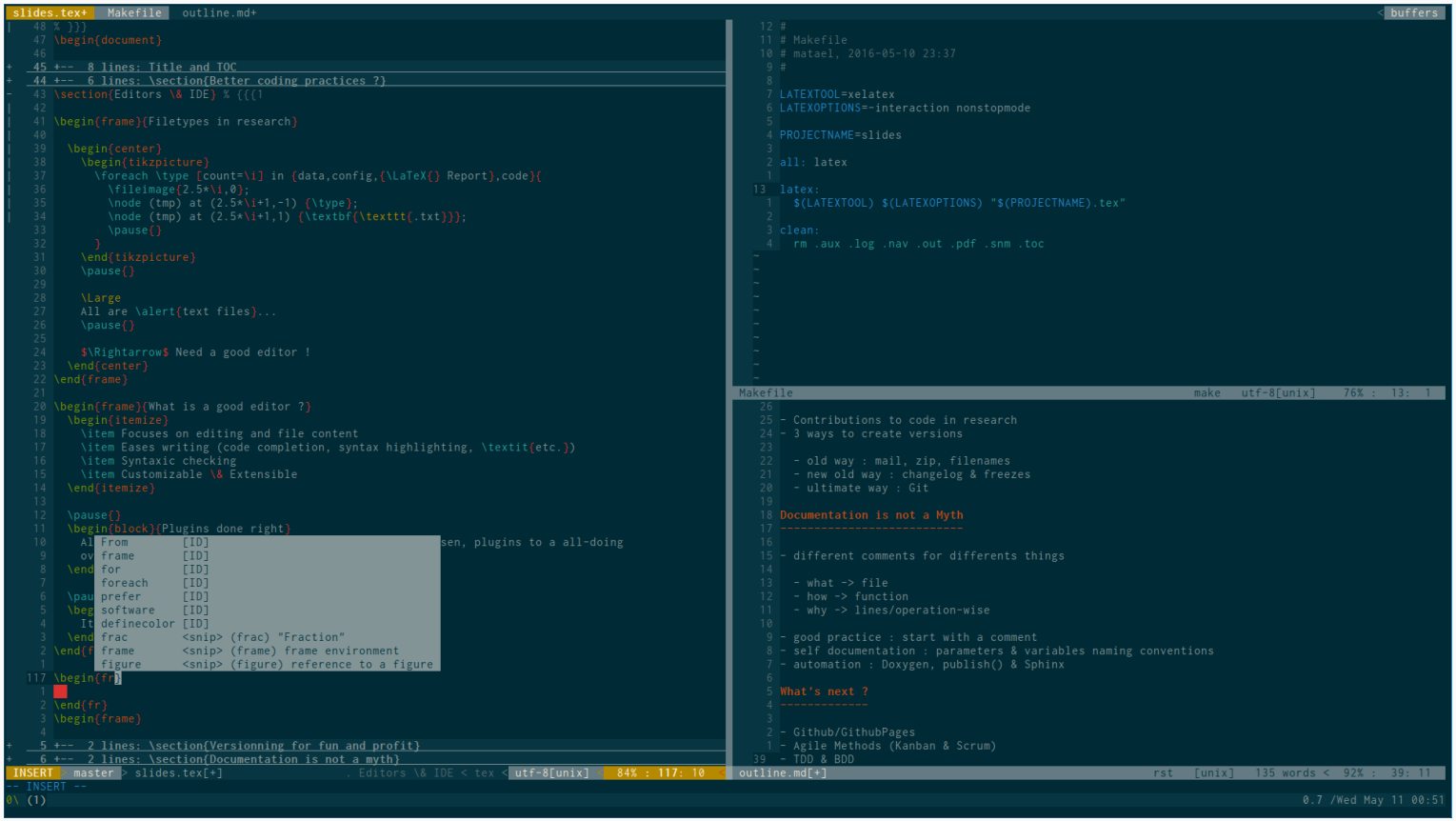
Editors: VIm

What people think it is...



Editors: VIm

What it does actually look like



Editors: VIm

Characteristics

- Command line or GUI
- Modal (one mode per class of action: insertion, displacement, selection, etc...)
- Based on shortcuts and commands
- Fully customizable through plugins or embedded language
- Painless integration with any standard UNIX toolchain

Where to start

- `http://vim-adventures.com/` & `$ vimtutor`
- `http://vimcasts.org/`
- `https://vimebook.com/en`

Editors: for the fainthearted

Other graphical, multi-platform, open-source and actively maintained editors exist

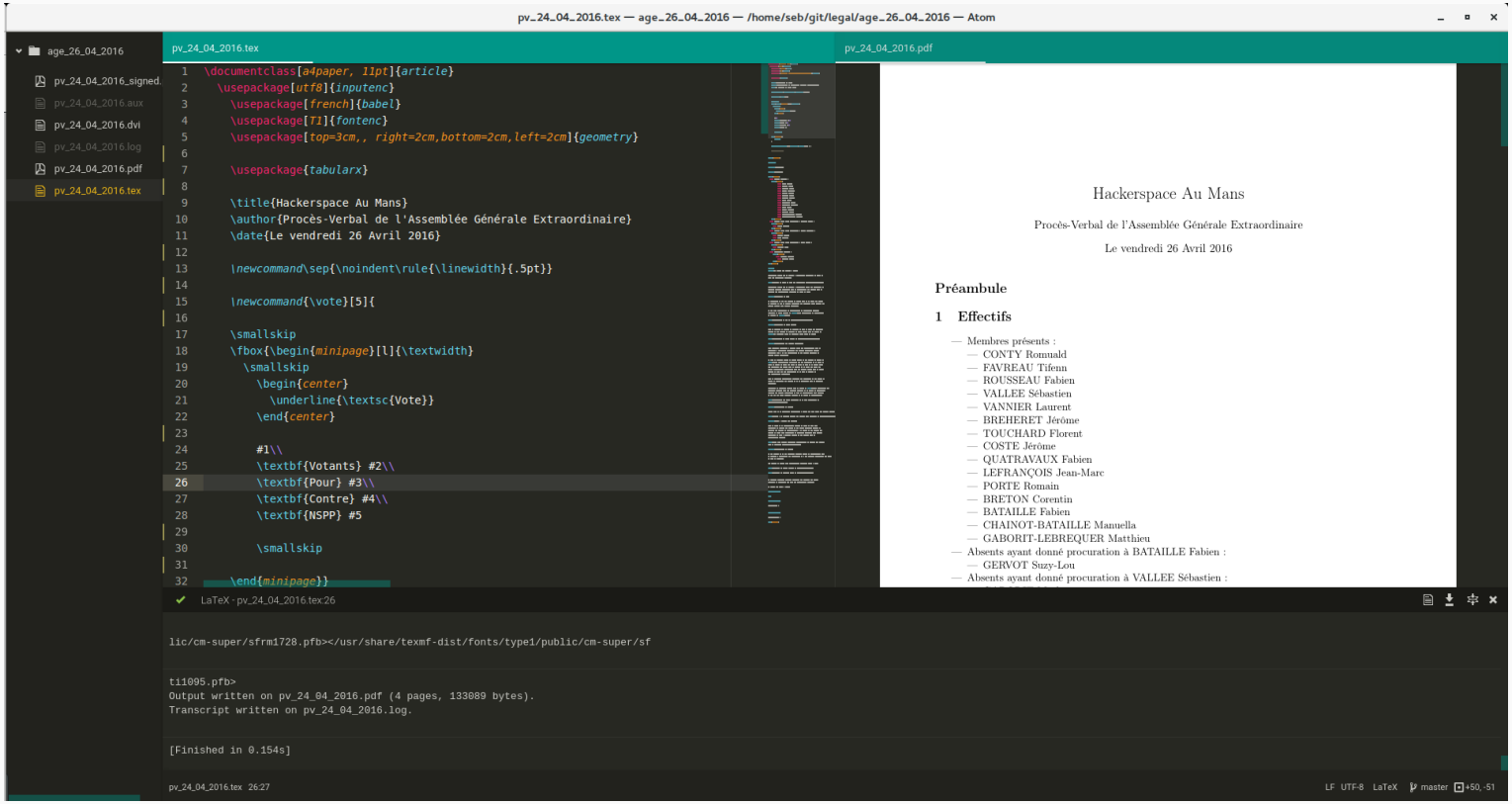
- **Atom** `atom.io`
- **Light Table** `lighttable.com`
- **Sublime Text** `sublimetext.com`

About Atom

- Developed by Github
- Lots of plugins & personalisation features
- Lightweight and easy to use
- Integration with other tools

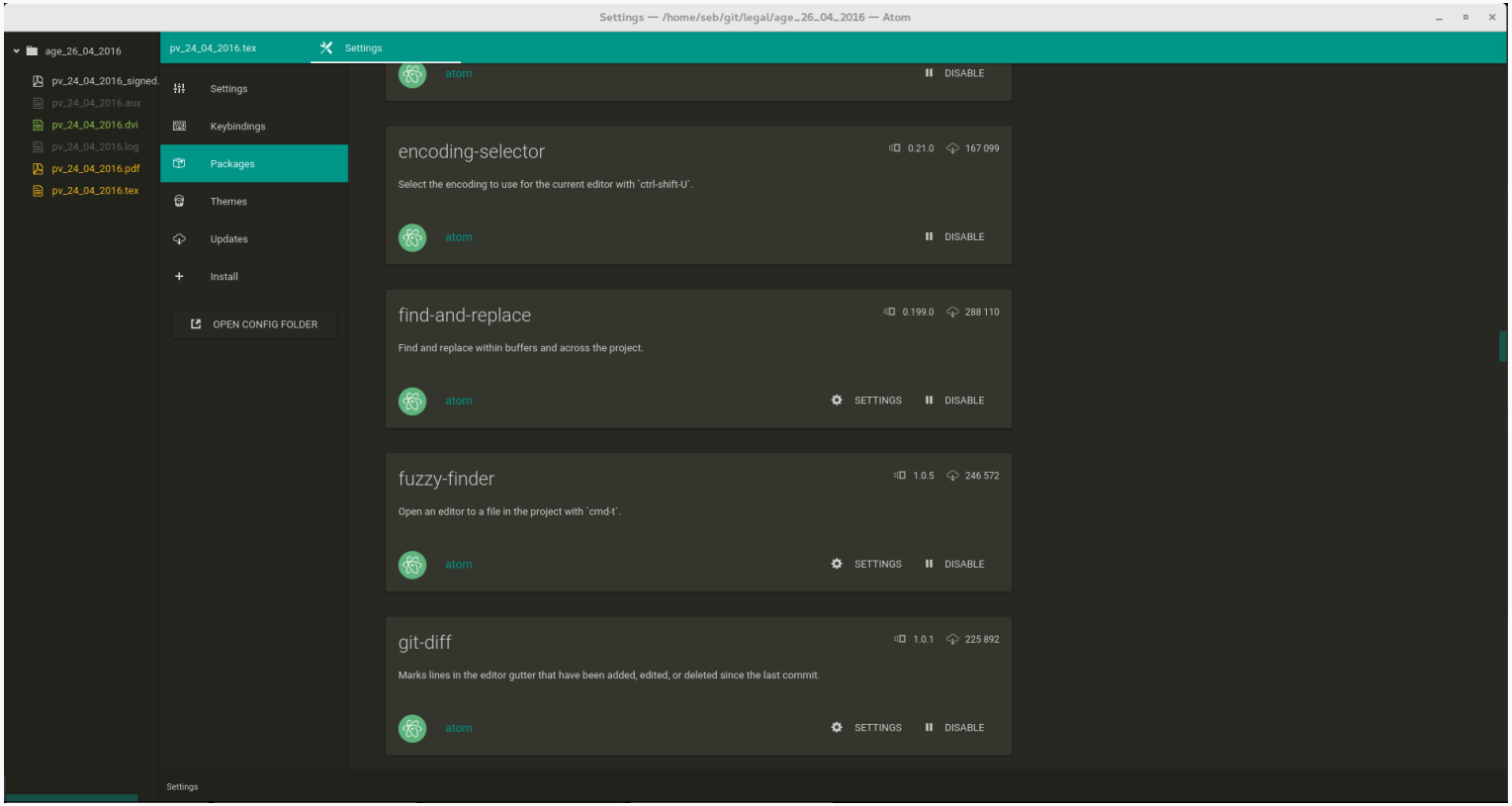
Editors: Atom

Integrated with other tools (IDE)



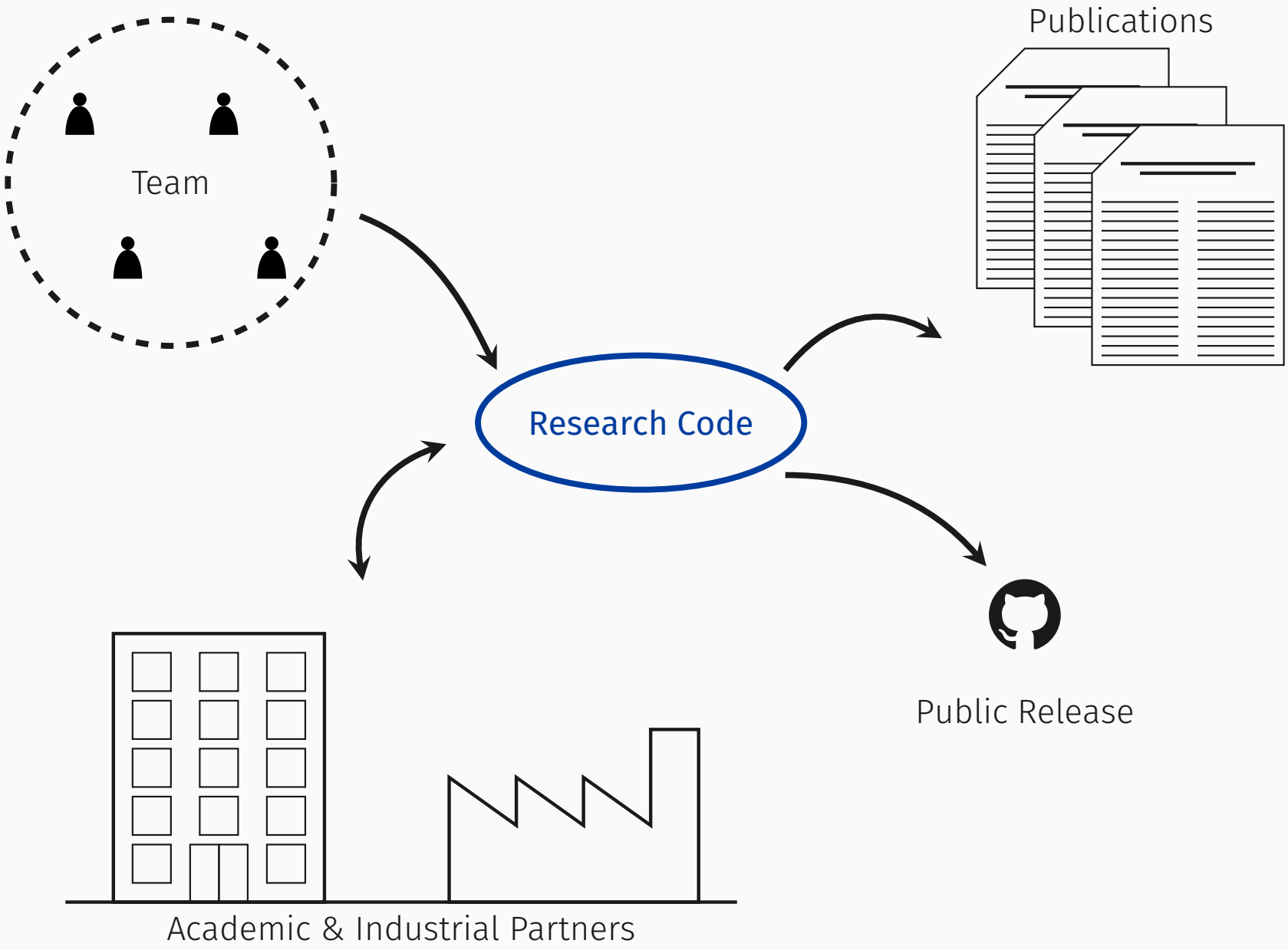
Editors: Atom

Built-in plugin manager



Versioning for fun and profit

Contribution to a research code



Versioning

We need to edit files every single day.

Versioning

We need to collaborate on files every single day.

Versioning

We need to review files every single day.

Versioning

We need to **explore** files every single day.

Versioning

We need to share files every single day.

Versioning

We need a reliable versioning system!

Versioning for fun and profit

The old old way : name it, zip it, mail it

Versioning: the old old way

The first versioning attempts traditionally rely on 3 pillars:

- Naming conventions
- Regular backups
- Careful data management

Versioning: the old old way

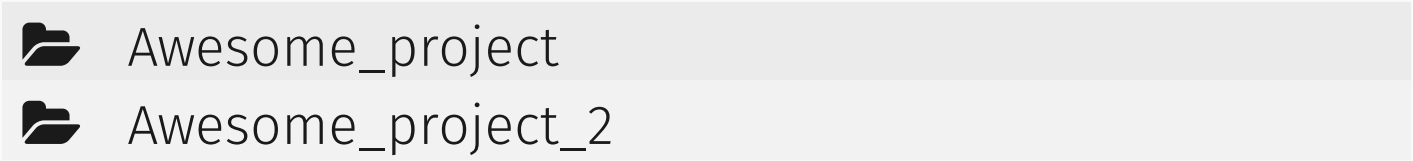
The first versioning attempts traditionally rely on 3 pillars:

- Naming conventions
- Regular backups
- Careful data management

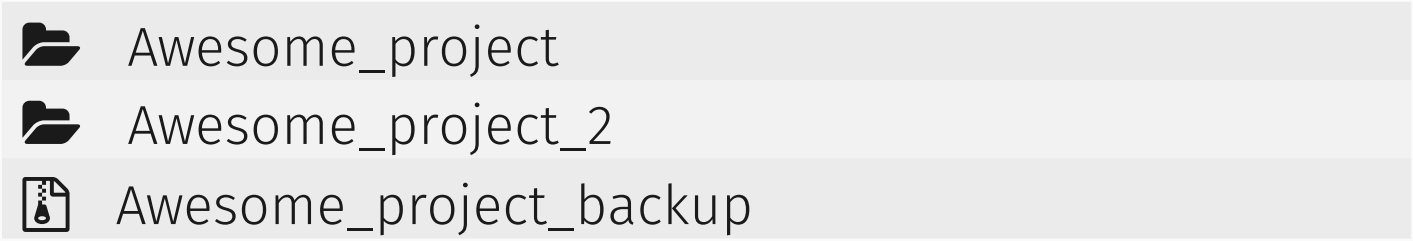
"C'mon, I even mailed myself the latest version: how could this go wrong?"

Versioning: the old old way

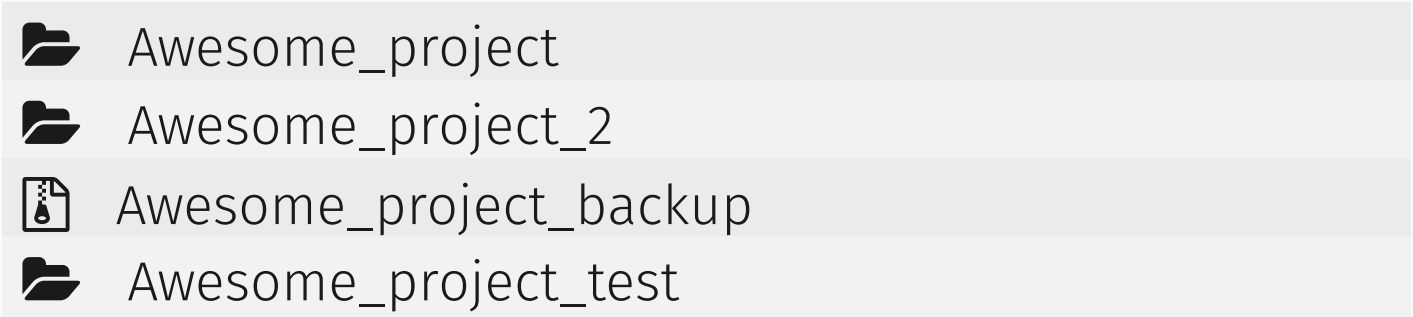
Versioning: the old old way



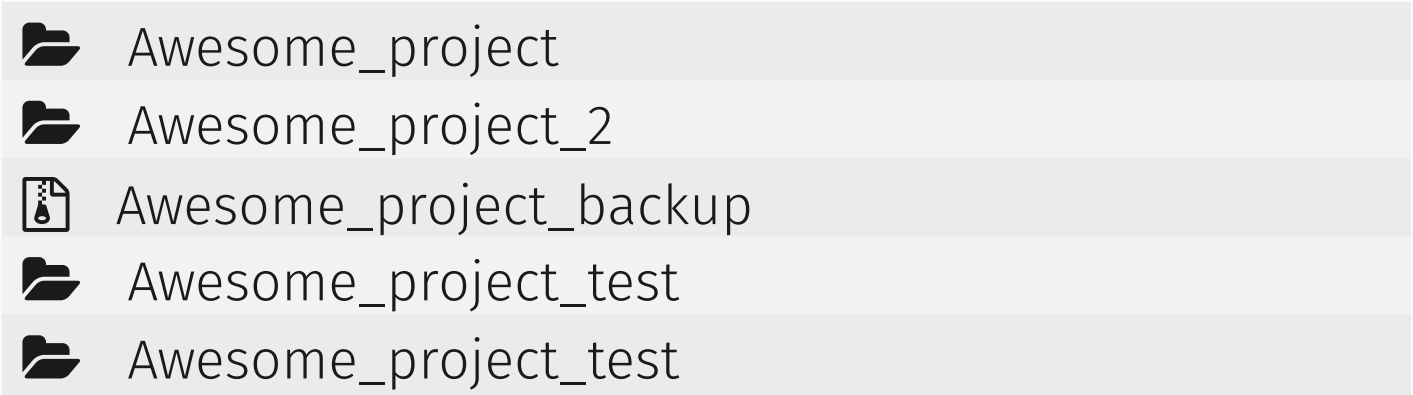
Versioning: the old old way



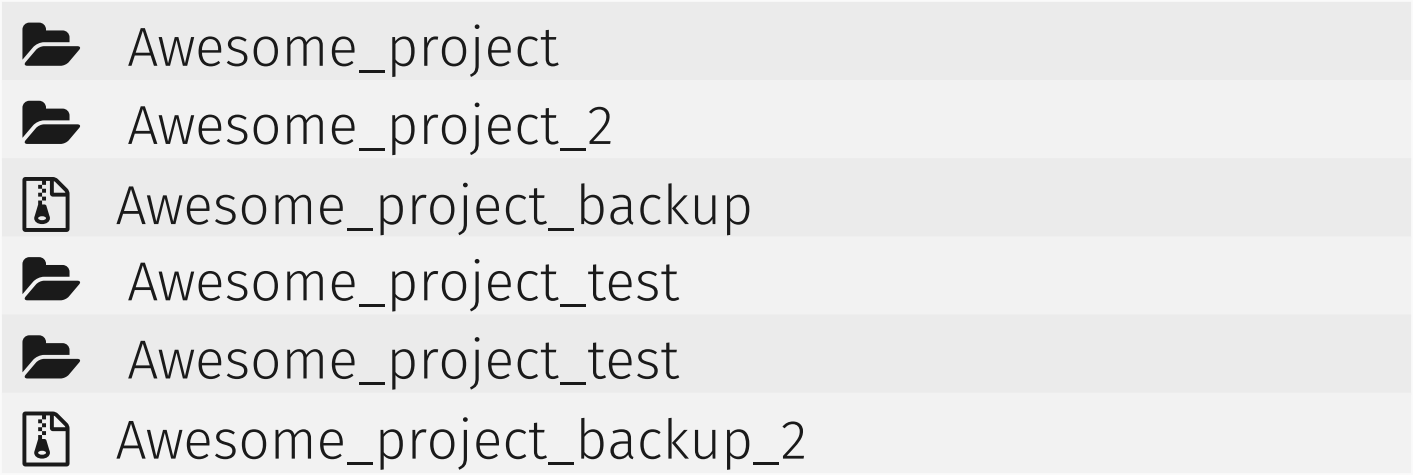
Versioning: the old old way



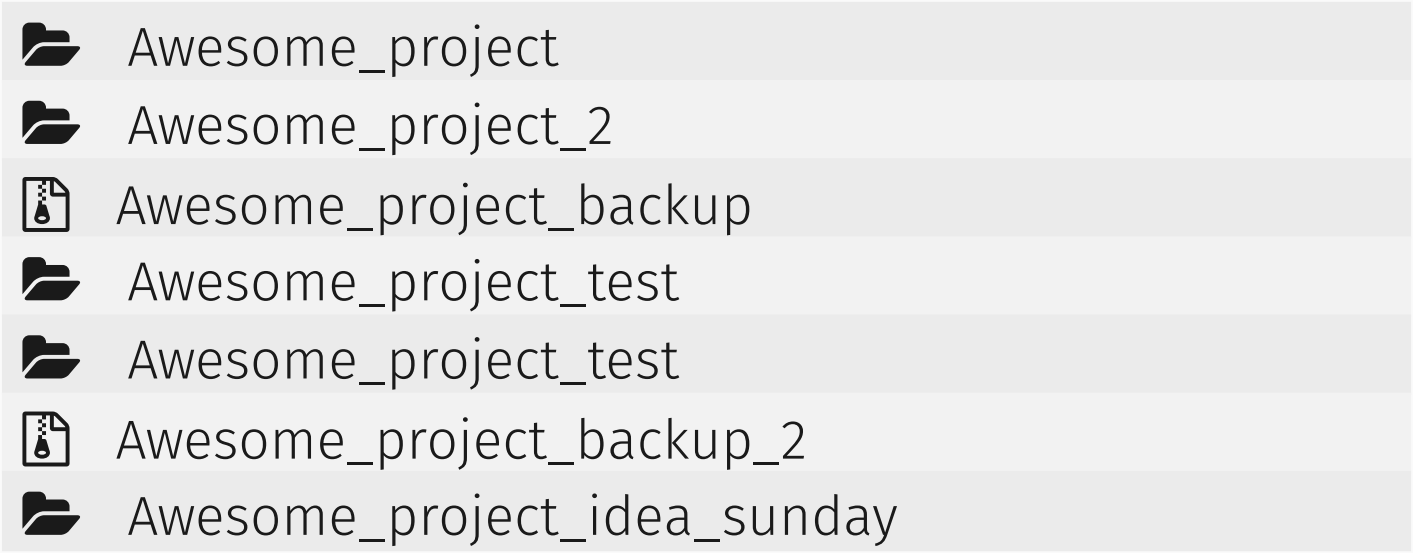
Versioning: the old old way



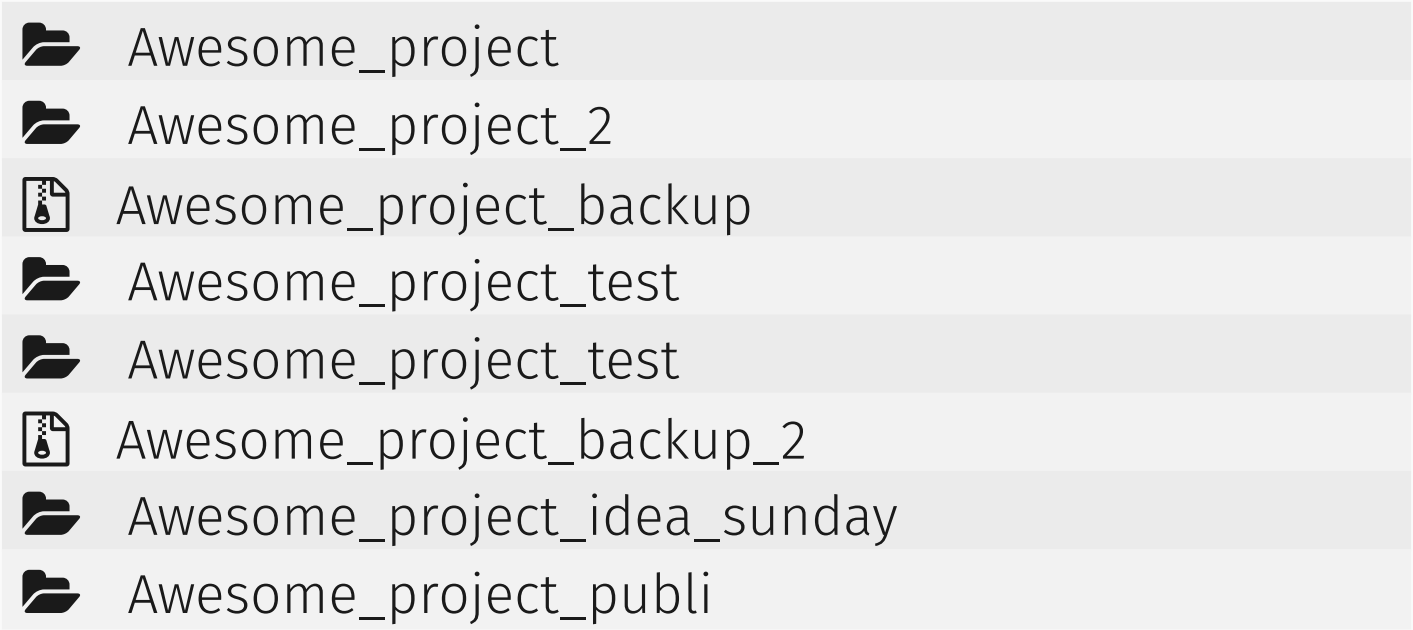
Versioning: the old old way



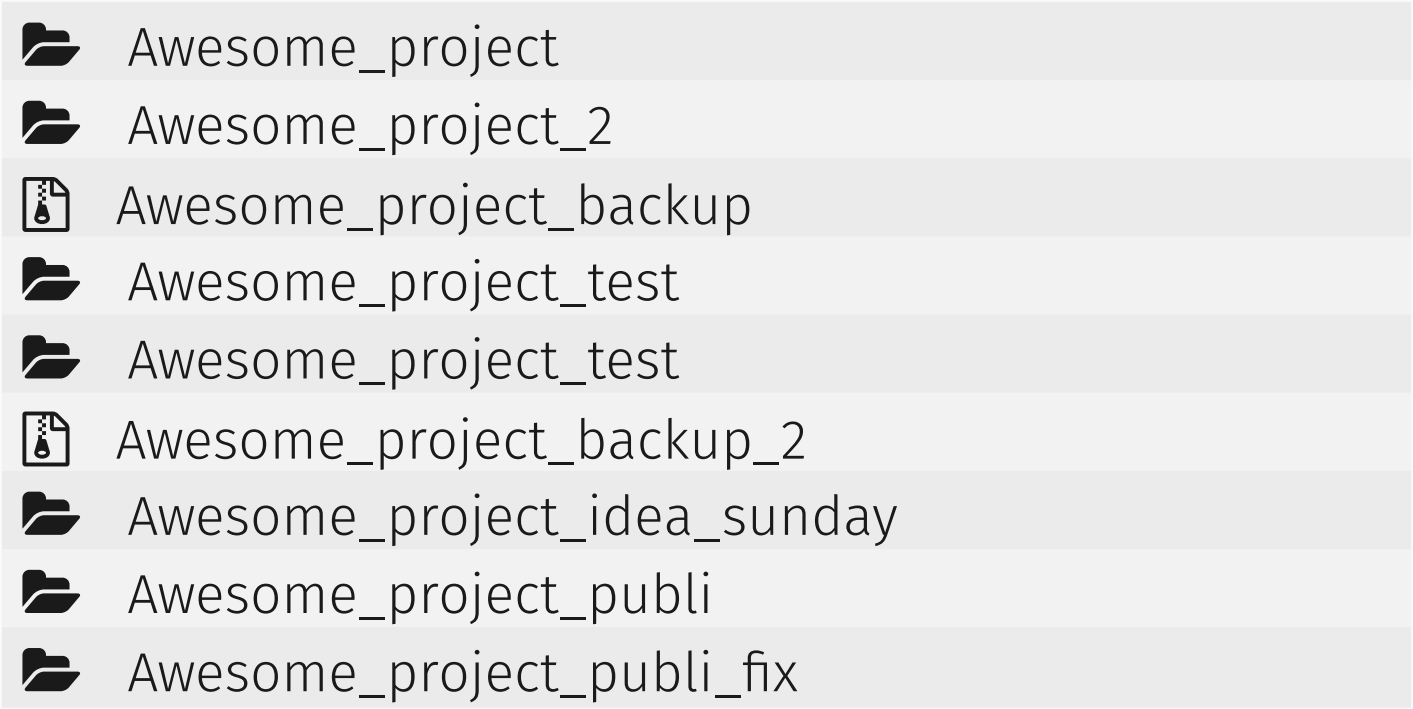
Versioning: the old old way



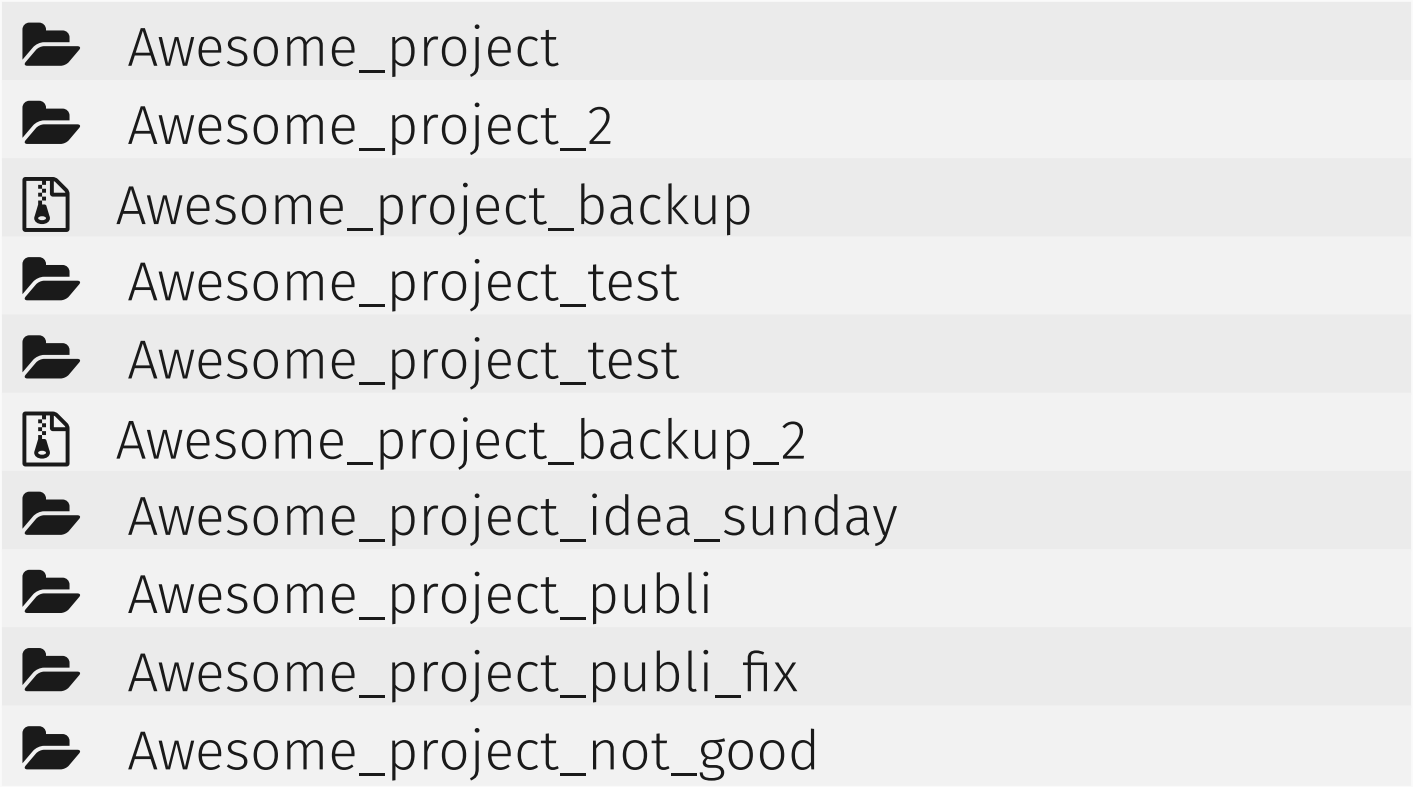
Versioning: the old old way



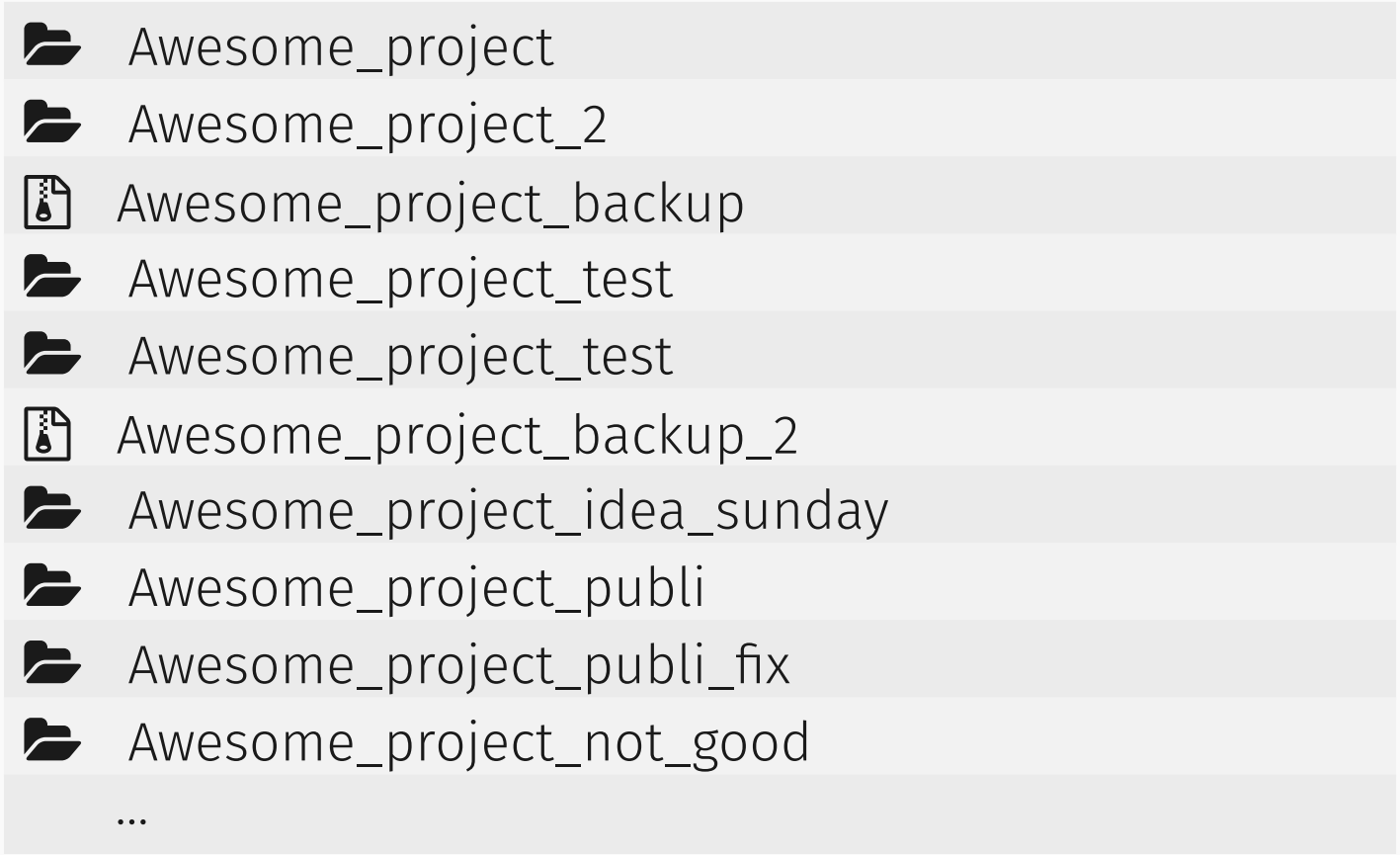
Versioning: the old old way



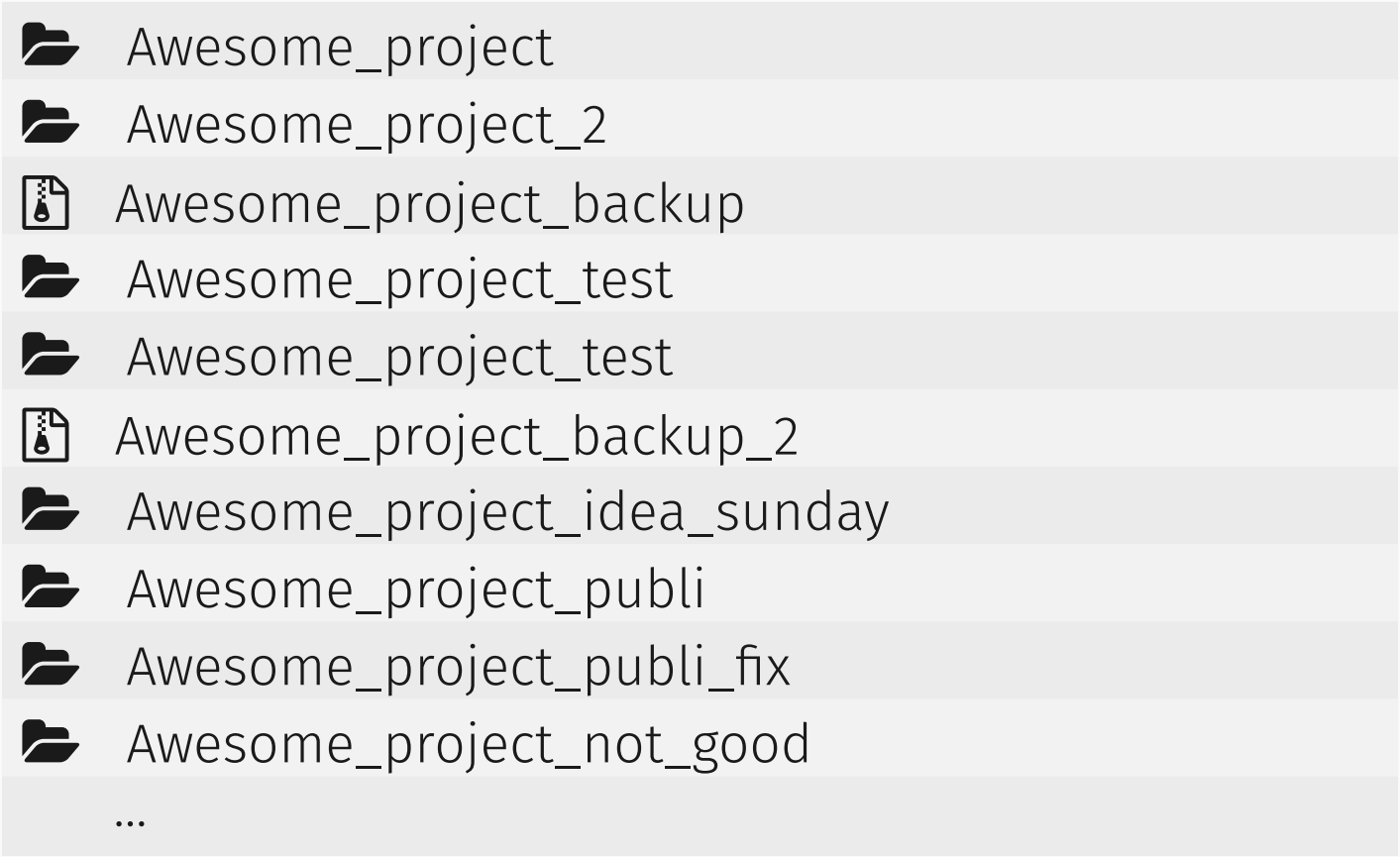
Versioning: the old old way



Versioning: the old old way



Versioning: the old old way



Versioning: the old old way

Plus, this does not behave well when scaling things up...

- How to handle team-work?
- Modification conflicts: what to do?
- No continuous time travel
- ...and a lot more issues

Versioning for fun and profit

The new old way : {Drop,}Box and Changelogs

Versioning: the new old way

The Cloud: one more step towards real-time collaboration!

- Free & paid plans
- Auto-sync even for big files
- Team notifications
- Mobile access to files
- A carefully maintained changelog *may* keep track of changes

Versioning: the new old way

The Cloud: one more step towards real-time collaboration!

- Free & paid plans
- Auto-sync even for big files
- Team notifications
- Mobile access to files
- A carefully maintained changelog *may* keep track of changes

But...

Versioning: the new old way

The Cloud: one more step towards real-time collaboration!

- Free & paid plans
- Auto-sync even for big files
- Team notifications
- Mobile access to files
- A carefully maintained changelog *may* keep track of changes

But... does the ”Cloud” really share the spirit that drives research?

Terms of Service: chosen moments

These and other features may require our systems to access, store and scan Your Stuff.

— DropBox TOS

Terms of Service: chosen moments

*You hereby grant Box and its contractors the right to **transmit, use and disclose** Content posted on the Service solely to the extent necessary to provide the Service, as otherwise permitted by these Terms, or to **comply with any request** of a governmental or regulatory body (including subpoenas or court orders)*

— Box TOS

Terms of Service: chosen moments

When you upload, submit, store, send or receive content to or through our Services, you give Google (and those we work with) a worldwide license to ***use, host, store, reproduce, modify, create derivative works, communicate, publish, publicly perform, publicly display and distribute*** *such content. [...] This license continues even if you stop using our Services [...]*

— *Google TOS*

Terms of Service: chosen moments

What about iCloud?

> *Apple reserves the right at all times to determine whether Content is appropriate and in compliance with this Agreement, and may pre-screen,* ***move, refuse, modify and/or remove*** *Content at any time, without prior notice and in its sole discretion [...]*
>
> — *iCloud TOS*

Terms of Service: chosen moments

What about iCloud?

> *However, by submitting or posting such Content on areas of the Service that are accessible by the public or other users with whom you consent to share such Content, you grant Apple a worldwide, royalty-free, non-exclusive license to* ***use, distribute, reproduce, modify, adapt, publish, translate, publicly perform and publicly display*** *such Content on the Service solely for the purpose for which such Content was submitted or made available, without any compensation or obligation to you.*
>
> — *iCloud TOS*

Versioning for fun and profit

The badass way: **git**

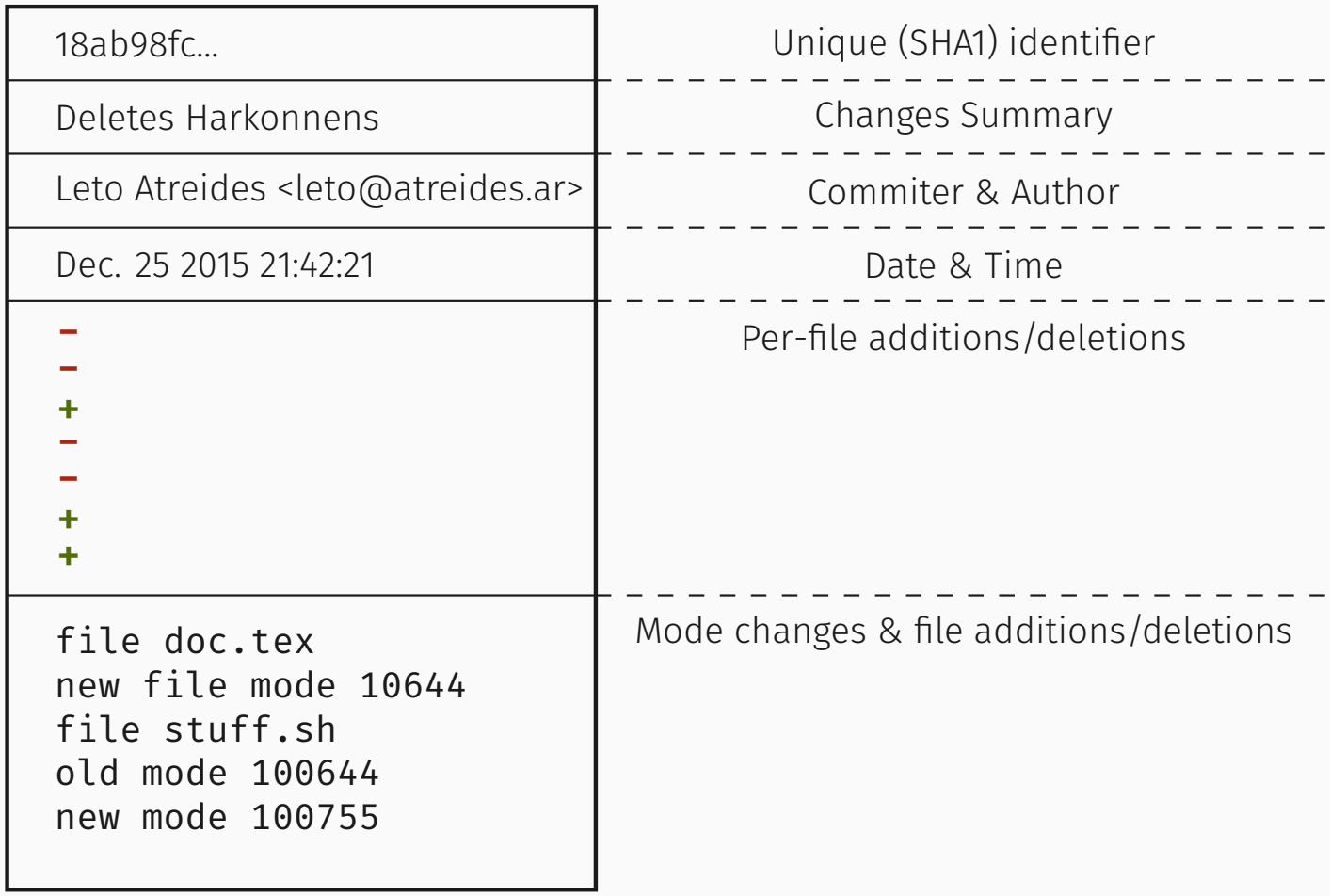
Versioning: the badass way

git was built to handle huge projects, like the Linux kernel:

- distributed (everyone has a full copy)
- efficient (only changes are transferred)
- commit-wise versioning
- supporting branches (new ideas in a enclosed space) and tagging
- potentially server-less & self-hostable
- simultaneous modifications & smart merging
- usable through GUI (tig, gitk, tortoise-git & others) or command-line

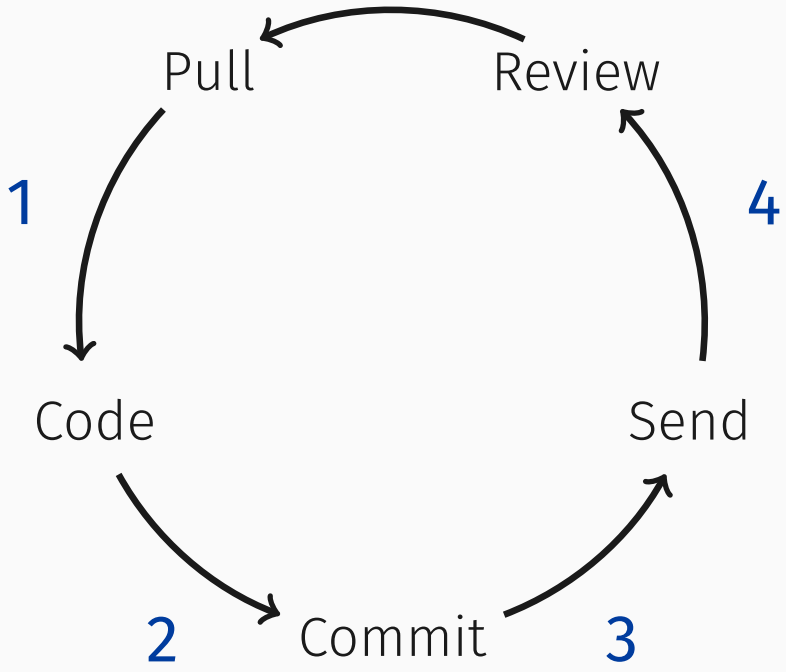
Versioning: the badass way

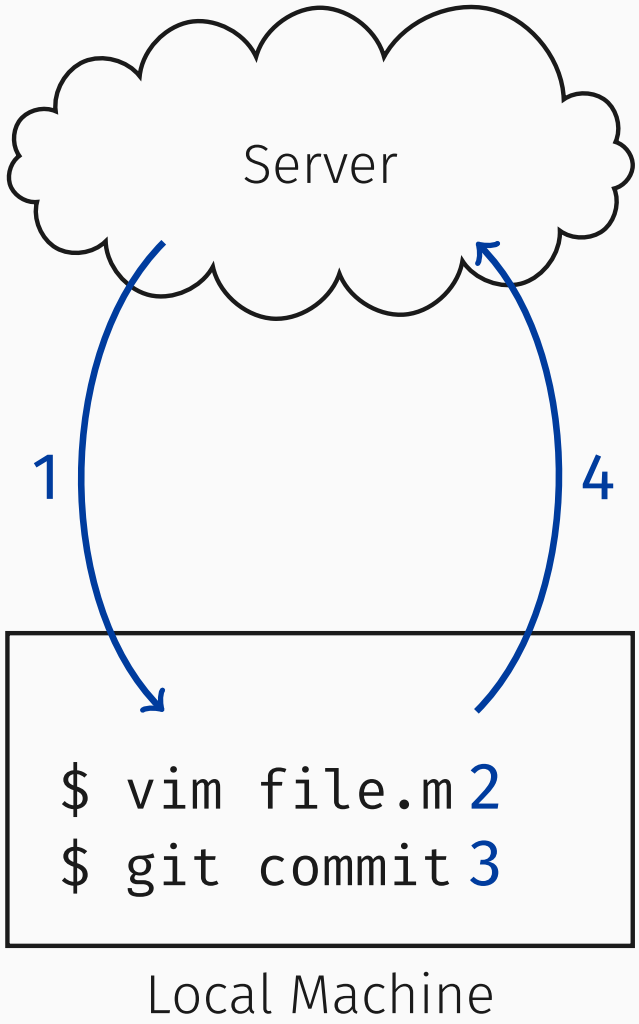
Concerning commits...



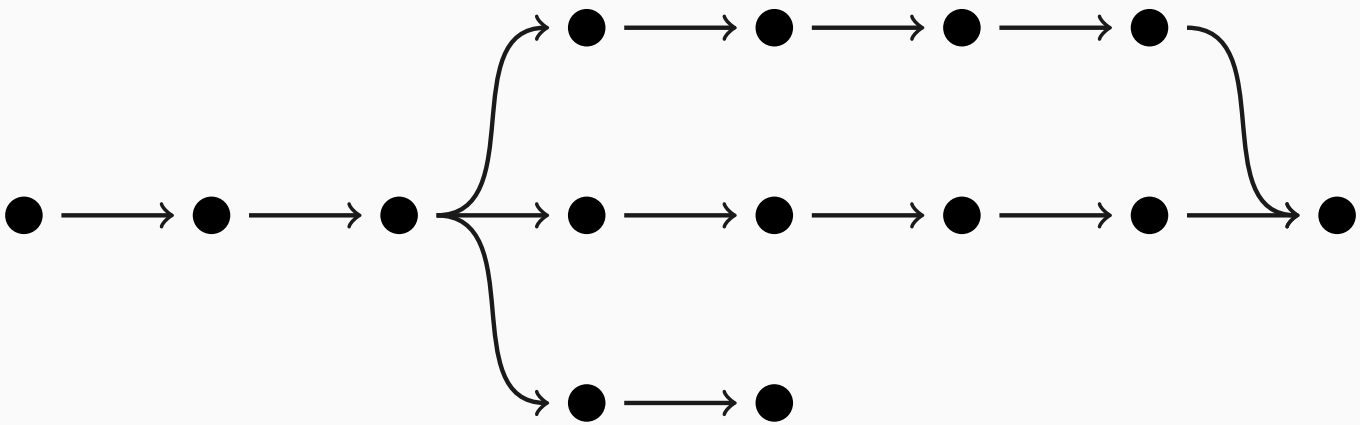
All the needed information is here!

Versioning: the badass way

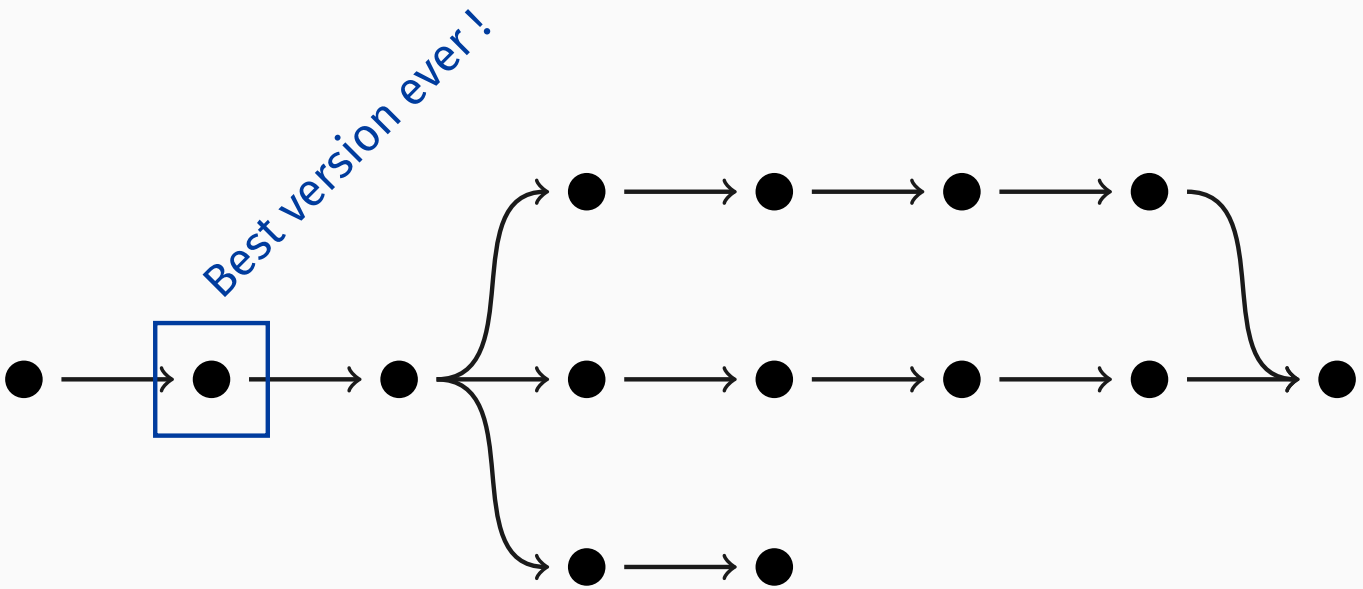




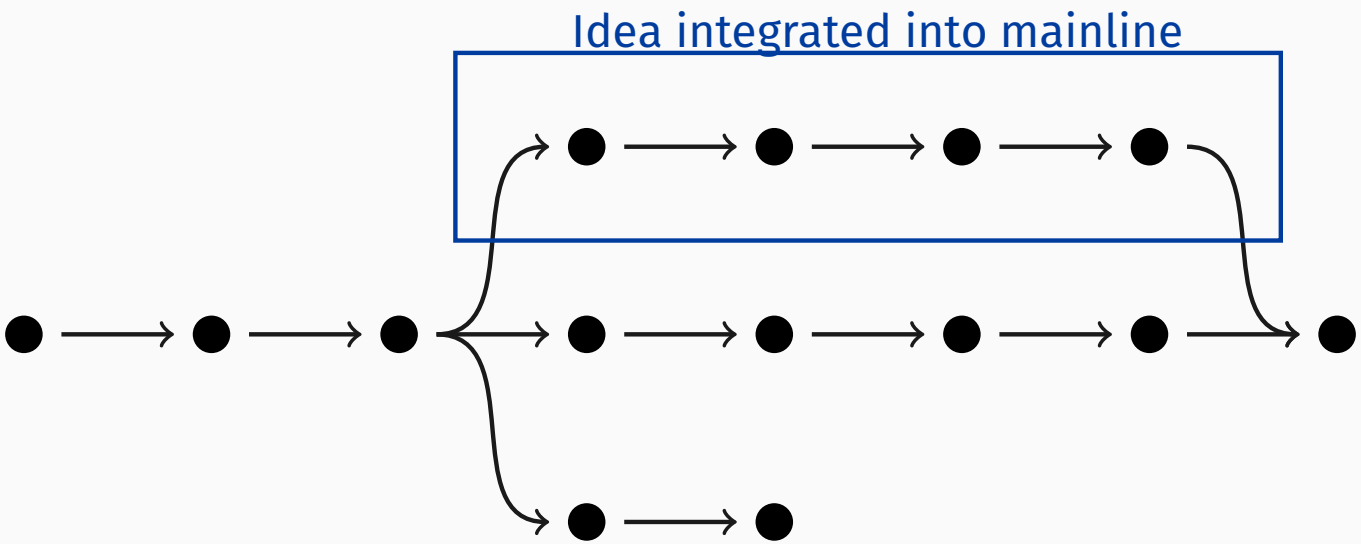
Versioning: the badass way



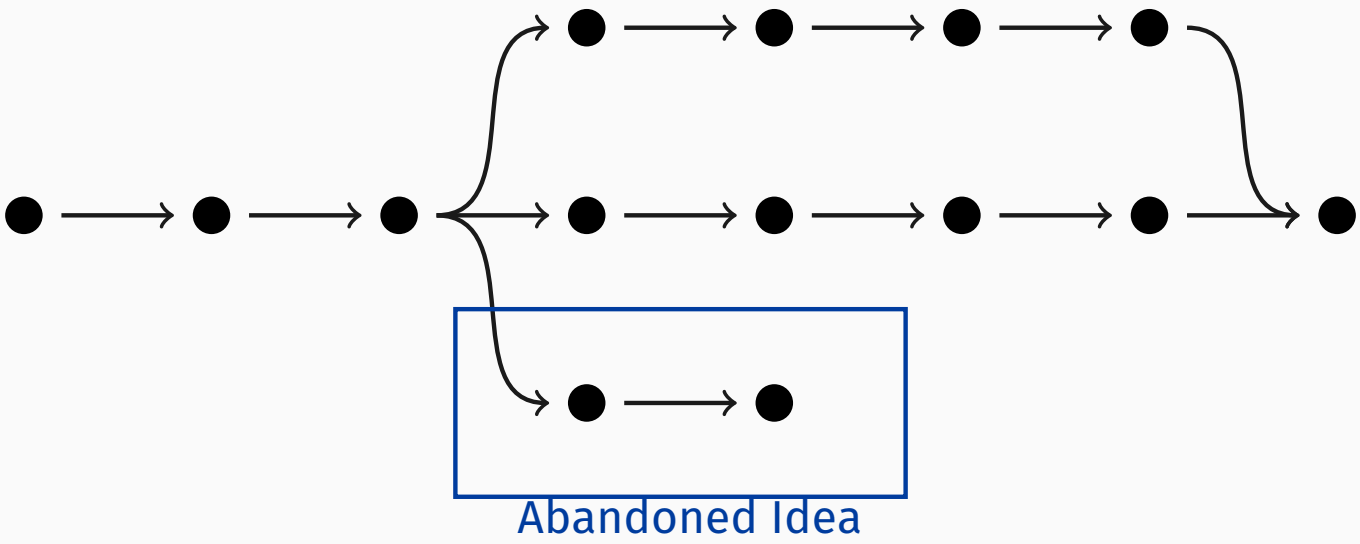
Versioning: the badass way



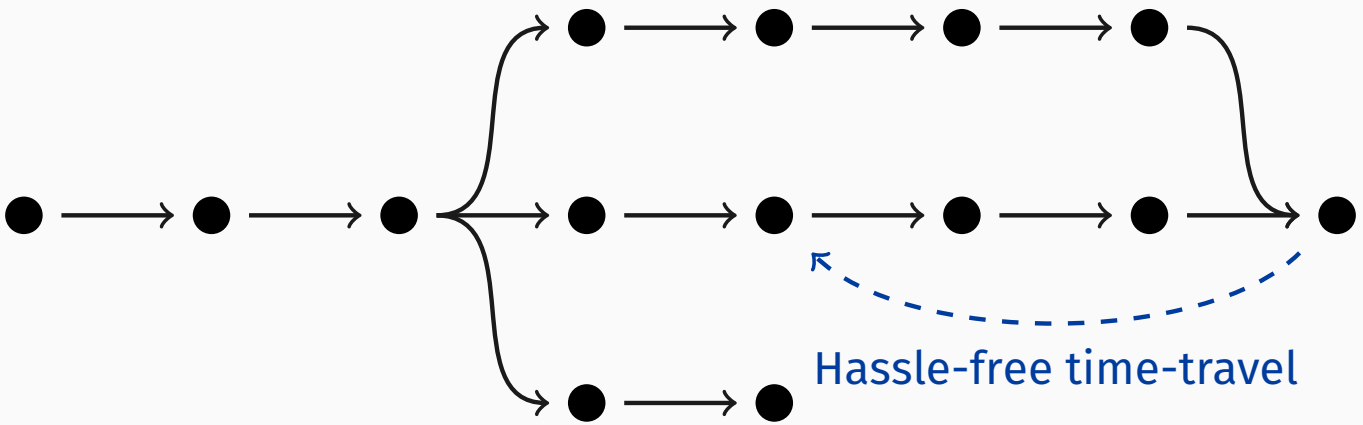
Versioning: the badass way



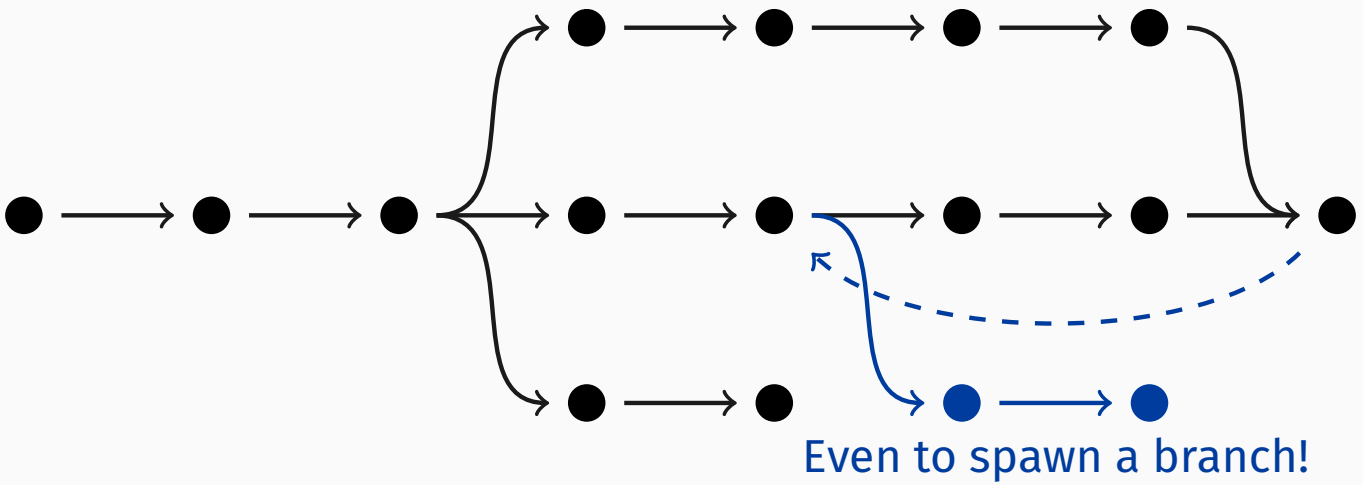
Versioning: the badass way



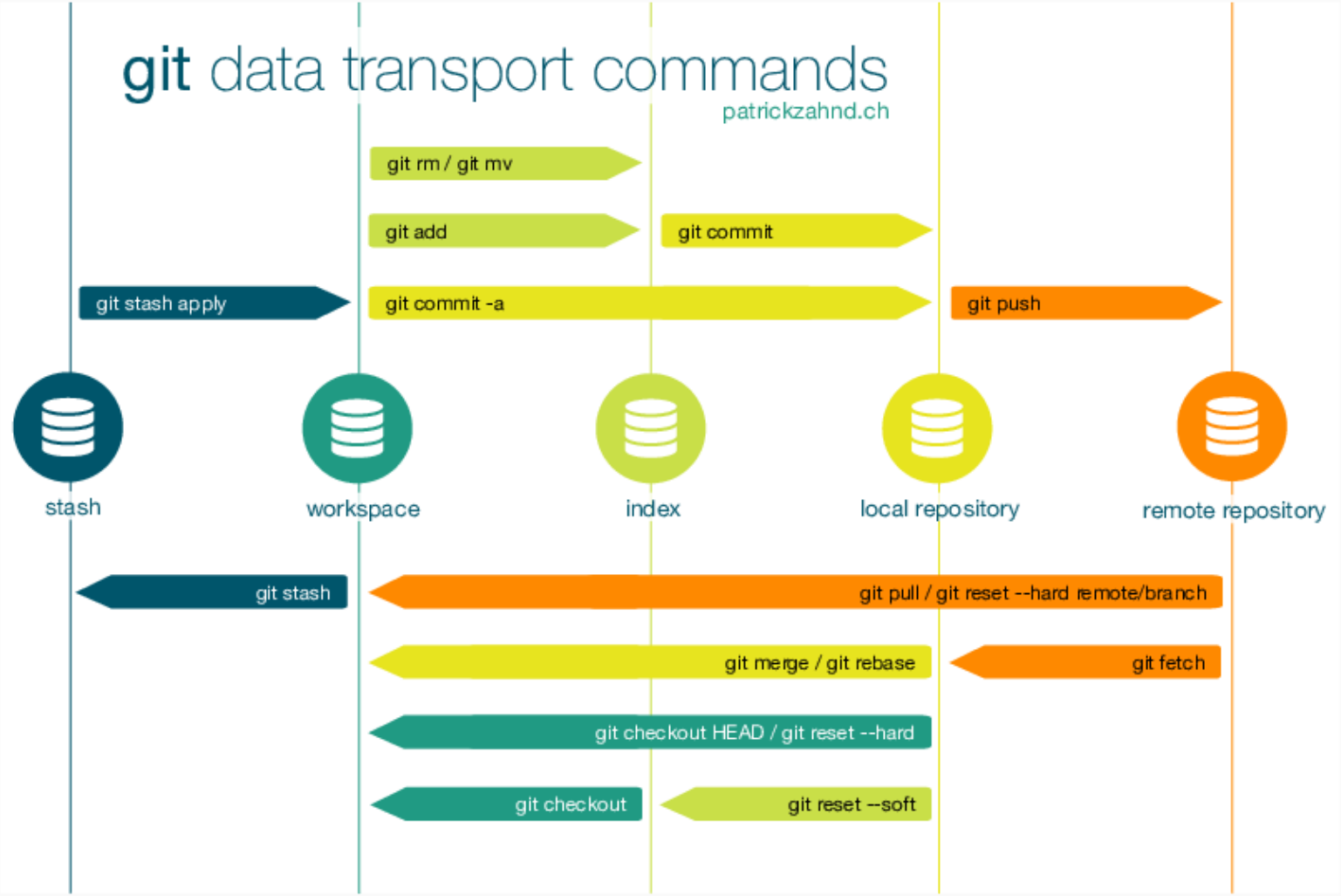
Versioning: the badass way



Versioning: the badass way



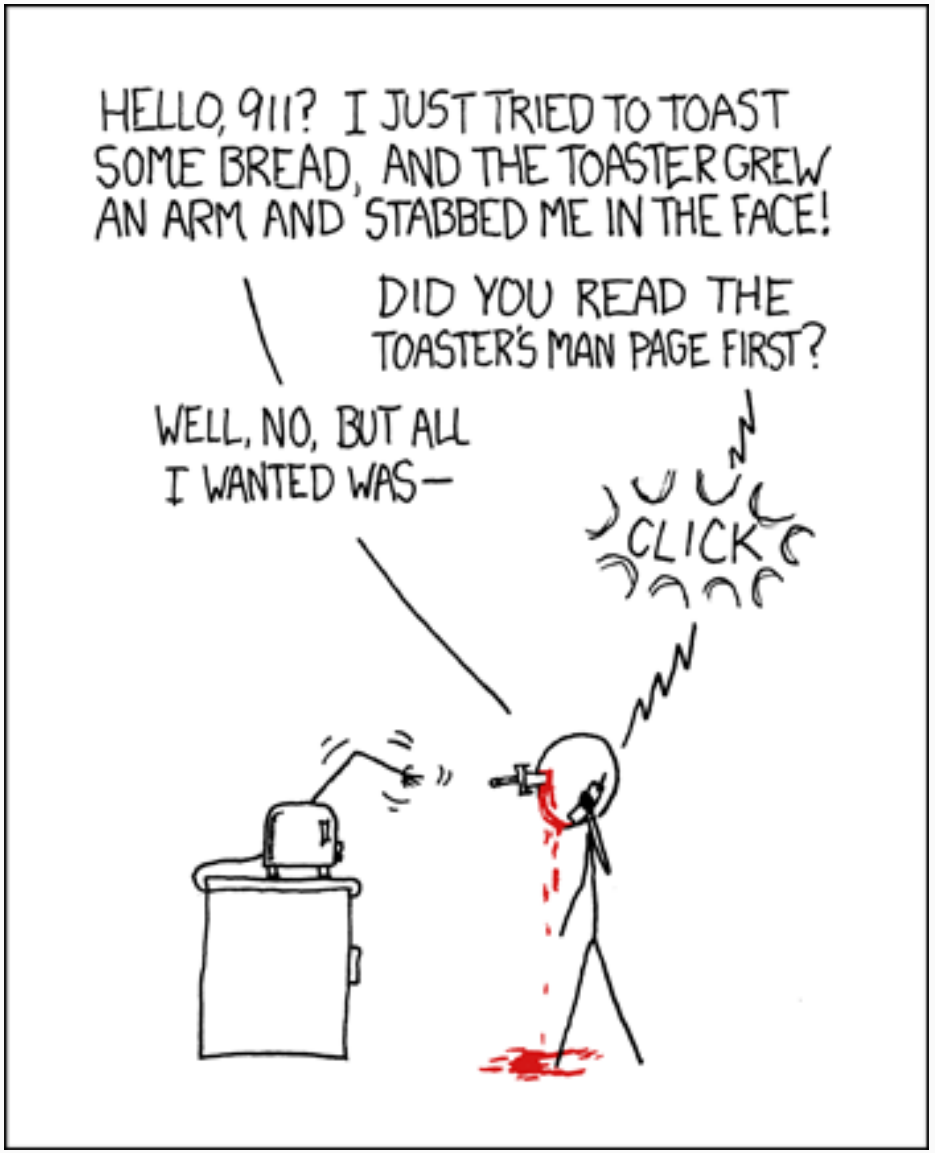
Versioning: the badass way



huit.re/gitstages

Documentation

Documentation



Courtesy of Randall Munroe, https://xkcd.com/293

Documentation: why

Before using a program, it's good to know:

- **what** it does
- **how** it does it
- **why** it does it like that

Documentation: why

Before using a program, it's good to know:

- **what** it does
- **how** it does it
- **why** it does it like that

Code tells you **how**...

...documentation tells you the rest.

Documentation: types

Internal

- comments
- self-documenting code

External

- separate manual
- referenced material

Documentation

Internal documentation

Internal documentation: comments

Begin every file with a comment

- say what the code should do
- describe how to use the file

A MATLAB example

```
function [h,a] = pythagoras(c1,c2)
% pythagoras -- computes the hypotenuse and the area of
%               right-angled triangle
%
% Inputs: c1 - length of the first cathetus [m]
%         c2 - length of the second cathetus [m]
% Outputs: h - length of the hypotenuse [m]
%          a - area of the triangle [m^2]
%
% Example: [hypo,area] = pythagoras(3,4)
```

Internal documentation: comments

- Explain **why**
- Close to code they refer to
- If updating the code, **update the comments**
- Avoid them if unnecessary

Internal documentation: comments

What is “unnecessary”

What feels obvious to you when writing the code may not be obvious to someone else, or even to you at a future time.

A MATLAB example

```
%   ** USELESS COMMENT **
% compute the mean value of every column of A
B = mean(A,1);

%   ** USEFUL, bsxfun may be obscure! **
% divide every column of A by the column vector v
B = bsxfun(@rdivide,A,v);
```

Internal documentation: comments

When commenting...

- ...a library, say what it does

```
% This toolbox computes PSDs.
```

- ...a function, say how it does it

```
% PSD estimate using Welch's method.
```

- ...a line, say why it does it that way

```
% bsxfun is faster than a for loop
```

Internal documentation: self-documenting code

> *Programs must be written for people to read, and only incidentally for machines to execute.*
>
> — *Structure and interpretation of computer programs*

- Aim at writing code that does not need comments
- Use meaningful, non-ambiguous variable names

Documentation

External documentation

External documentation

Quality internal documentation can be used to automatically generate external documentation.

A MATLAB example

publish() code publishing tool included in MATLAB

More useful tools

Doxygen doc-generator supporting multiple languages

Sphinx doc-generator built for the Python project

rtfd.org free web-service for hosting documentation

What’s next?

Cut scenes

We couldn't fit it all in... so here is the rest:

Github Public Git server for free software (paid plans exist) with webpage publishing service (good to host the homepage of a project).

Agility Project management methods centered on human relations more than on tools and processes. See Kanban or Scrum for an introduction.

TDD & BDD Test- & Behavior Driven Development. Coding paradigms focusing on certain aspects (reliability & spec. compliance). Useful to scale from a research PoC to a distributed software.

References

Slides

github.com/KTH/MWL_LunchSeminar_CodingToolsAndPractices

Editors Cheatsheets

huit.re/emacscs & huit.re/vimcs

Versioning

huit.re/gitcs & huit.re/gitstages

Documentation

huit.re/zenofpython & huit.re/pep8

Questions?

manzari@kth.se — gaborit@kth.se