



# Coding Best Practices & Tools

Towards painless maintainable programming for researchers

---

L. MANZARI – M. GABORIT

MWL Lunch Seminar – June 2016

1. Better coding practices ?
2. Editors & IDE
3. Versionning for fun and profit
4. Documentation is not a myth
5. What's next ?

Better coding practices ?

---

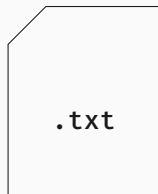
TODO

## Editors & IDE

---

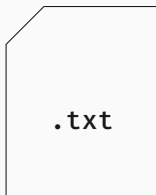
# Filetypes in research

# Filetypes in research

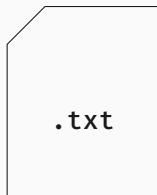


data

# Filetypes in research



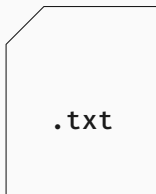
data



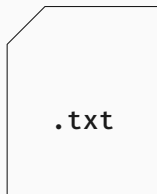
config



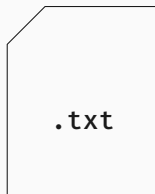
# Filetypes in research



data

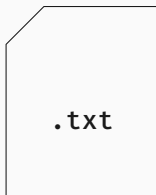


config

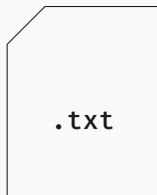


L<sup>A</sup>T<sub>E</sub>X Report

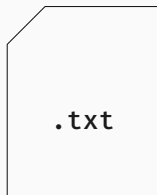
# Filetypes in research



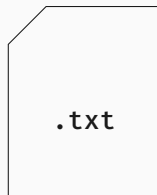
data



config

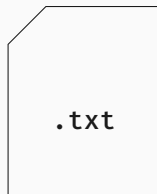


LaTeX Report

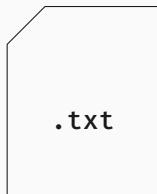


code

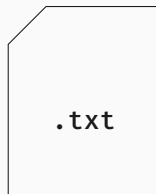
# Filetypes in research



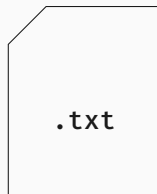
data



config



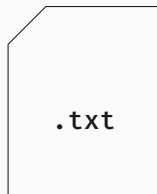
L<sup>A</sup>T<sub>E</sub>X Report



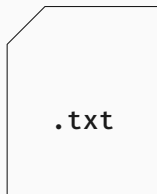
code

All are text files...

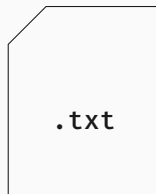
# Filetypes in research



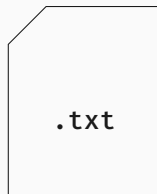
data



config



L<sup>A</sup>T<sub>E</sub>X Report



code

All are **text files**...

⇒ Need a good editor !

# What is a good editor ?

- Focuses on editing and file content
- Eases writing (code completion, syntax highlighting, etc.)
- Syntactic checking
- Customizable & Extensible

# What is a good editor ?

- Focuses on editing and file content
- Eases writing (code completion, syntax highlighting, *etc.*)
- Syntactic checking
- Customizable & Extensible

## **Plugins done right**

Always prefer a simple editor and a few, carefully chosen, plugins to a all-doing overcomplicated software.

# What is a good editor ?

- Focuses on editing and file content
- Eases writing (code completion, syntax highlighting, *etc.*)
- Syntactic checking
- Customizable & Extensible

## **Plugins done right**

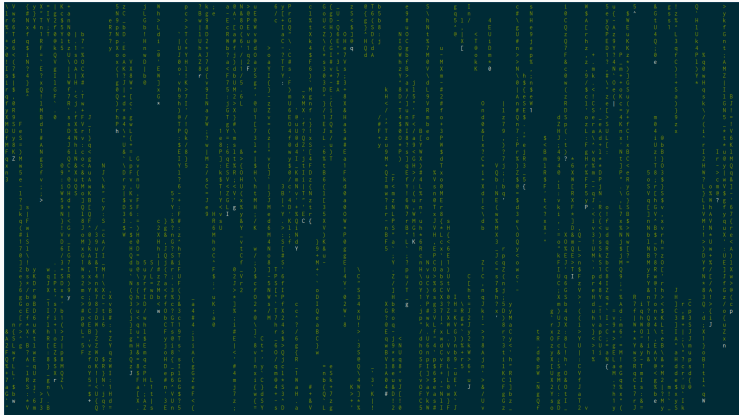
Always prefer a simple editor and a few, carefully chosen, plugins to a all-doing overcomplicated software.

## **Plugins done right (UNIX Version)**

It does only one thing but it does it perfectly.

# Editors : Vim

What people think it is...





Editors : VIm

## What it does actually look like

```

10 \begin{text}
11 \outlinetoc
12 \end{text}
13 \begin{document}
14
15 \section{Title and TOC}
16
17 \section{Introduction (Master coding practices 7)}
18
19 \section{Editors & IDEs}
20
21 \begin{frame}[Filetypes in research]
22
23 \begin{center}
24 \begin{tikzpicture}
25 \foreach \Name [count=\i] in {data,config,{\LaTeXe} Report}.code{
26 \fill[blue!2.5!white]
27 \node[cmpt] at (2.5*\i-1) {\Name};
28 \node[cmpt] at (2.5*\i+1) {\Name(steeff,txt)};
29 \draw[>]
30 }
31 \end{tikzpicture}
32 \end{center}
33
34 \begin{frame}
35
36 \begin{center}
37 \begin{tikzpicture}
38 \draw[>]
39
40 \end{tikzpicture}
41 \end{center}
42
43 \begin{frame}
44
45 \begin{center}
46 \begin{tikzpicture}
47 \draw[>]
48
49 \end{tikzpicture}
50 \end{center}
51
52 \begin{frame}
53
54 \begin{center}
55 \begin{tikzpicture}
56 \draw[>]
57
58 \end{tikzpicture}
59 \end{center}
60
61 \begin{frame}
62
63 \begin{center}
64 \begin{tikzpicture}
65 \draw[>]
66
67 \end{tikzpicture}
68 \end{center}
69
70 \begin{frame}
71
72 \begin{center}
73 \begin{tikzpicture}
74 \draw[>]
75
76 \end{tikzpicture}
77 \end{center}
78
79 \begin{frame}
80
81 \begin{center}
82 \begin{tikzpicture}
83 \draw[>]
84
85 \end{tikzpicture}
86 \end{center}
87
88 \begin{frame}
89
90 \begin{center}
91 \begin{tikzpicture}
92 \draw[>]
93
94 \end{tikzpicture}
95 \end{center}
96
97 \begin{frame}
98
99 \begin{center}
100 \begin{tikzpicture}
101 \draw[>]
102
103 \end{tikzpicture}
104 \end{center}
105
106 \begin{frame}
107
108 \begin{center}
109 \begin{tikzpicture}
110 \draw[>]
111
112 \end{tikzpicture}
113 \end{center}
114
115 \begin{frame}
116
117 \begin{center}
118 \begin{tikzpicture}
119 \draw[>]
120
121 \end{tikzpicture}
122 \end{center}
123
124 \begin{frame}
125
126 \begin{center}
127 \begin{tikzpicture}
128 \draw[>]
129
130 \end{tikzpicture}
131 \end{center}
132
133 \begin{frame}
134
135 \begin{center}
136 \begin{tikzpicture}
137 \draw[>]
138
139 \end{tikzpicture}
140 \end{center}
141
142 \begin{frame}
143
144 \begin{center}
145 \begin{tikzpicture}
146 \draw[>]
147
148 \end{tikzpicture}
149 \end{center}
150
151 \begin{frame}
152
153 \begin{center}
154 \begin{tikzpicture}
155 \draw[>]
156
157 \end{tikzpicture}
158 \end{center}
159
160 \begin{frame}
161
162 \begin{center}
163 \begin{tikzpicture}
164 \draw[>]
165
166 \end{tikzpicture}
167 \end{center}
168
169 \begin{frame}
170
171 \begin{center}
172 \begin{tikzpicture}
173 \draw[>]
174
175 \end{tikzpicture}
176 \end{center}
177
178 \begin{frame}
179
180 \begin{center}
181 \begin{tikzpicture}
182 \draw[>]
183
184 \end{tikzpicture}
185 \end{center}
186
187 \begin{frame}
188
189 \begin{center}
190 \begin{tikzpicture}
191 \draw[>]
192
193 \end{tikzpicture}
194 \end{center}
195
196 \begin{frame}
197
198 \begin{center}
199 \begin{tikzpicture}
200 \draw[>]
201
202 \end{tikzpicture}
203 \end{center}
204
205 \begin{frame}
206
207 \begin{center}
208 \begin{tikzpicture}
209 \draw[>]
210
211 \end{tikzpicture}
212 \end{center}
213
214 \begin{frame}
215
216 \begin{center}
217 \begin{tikzpicture}
218 \draw[>]
219
220 \end{tikzpicture}
221 \end{center}
222
223 \begin{frame}
224
225 \begin{center}
226 \begin{tikzpicture}
227 \draw[>]
228
229 \end{tikzpicture}
230 \end{center}
231
232 \begin{frame}
233
234 \begin{center}
235 \begin{tikzpicture}
236 \draw[>]
237
238 \end{tikzpicture}
239 \end{center}
240
241 \begin{frame}
242
243 \begin{center}
244 \begin{tikzpicture}
245 \draw[>]
246
247 \end{tikzpicture}
248 \end{center}
249
250 \begin{frame}
251
252 \begin{center}
253 \begin{tikzpicture}
254 \draw[>]
255
256 \end{tikzpicture}
257 \end{center}
258
259 \begin{frame}
260
261 \begin{center}
262 \begin{tikzpicture}
263 \draw[>]
264
265 \end{tikzpicture}
266 \end{center}
267
268 \begin{frame}
269
270 \begin{center}
271 \begin{tikzpicture}
272 \draw[>]
273
274 \end{tikzpicture}
275 \end{center}
276
277 \begin{frame}
278
279 \begin{center}
280 \begin{tikzpicture}
281 \draw[>]
282
283 \end{tikzpicture}
284 \end{center}
285
286 \begin{frame}
287
288 \begin{center}
289 \begin{tikzpicture}
290 \draw[>]
291
292 \end{tikzpicture}
293 \end{center}
294
295 \begin{frame}
296
297 \begin{center}
298 \begin{tikzpicture}
299 \draw[>]
300
301 \end{tikzpicture}
302 \end{center}
303
304 \begin{frame}
305
306 \begin{center}
307 \begin{tikzpicture}
308 \draw[>]
309
310 \end{tikzpicture}
311 \end{center}
312
313 \begin{frame}
314
315 \begin{center}
316 \begin{tikzpicture}
317 \draw[>]
318
319 \end{tikzpicture}
320 \end{center}
321
322 \begin{frame}
323
324 \begin{center}
325 \begin{tikzpicture}
326 \draw[>]
327
328 \end{tikzpicture}
329 \end{center}
330
331 \begin{frame}
332
333 \begin{center}
334 \begin{tikzpicture}
335 \draw[>]
336
337 \end{tikzpicture}
338 \end{center}
339
340 \begin{frame}
341
342 \begin{center}
343 \begin{tikzpicture}
344 \draw[>]
345
346 \end{tikzpicture}
347 \end{center}
348
349 \begin{frame}
350
351 \begin{center}
352 \begin{tikzpicture}
353 \draw[>]
354
355 \end{tikzpicture}
356 \end{center}
357
358 \begin{frame}
359
360 \begin{center}
361 \begin{tikzpicture}
362 \draw[>]
363
364 \end{tikzpicture}
365 \end{center}
366
367 \begin{frame}
368
369 \begin{center}
370 \begin{tikzpicture}
371 \draw[>]
372
373 \end{tikzpicture}
374 \end{center}
375
376 \begin{frame}
377
378 \begin{center}
379 \begin{tikzpicture}
380 \draw[>]
381
382 \end{tikzpicture}
383 \end{center}
384
385 \begin{frame}
386
387 \begin{center}
388 \begin{tikzpicture}
389 \draw[>]
390
391 \end{tikzpicture}
392 \end{center}
393
394 \begin{frame}
395
396 \begin{center}
397 \begin{tikzpicture}
398 \draw[>]
399
400 \end{tikzpicture}
401 \end{center}
402
403 \begin{frame}
404
405 \begin{center}
406 \begin{tikzpicture}
407 \draw[>]
408
409 \end{tikzpicture}
410 \end{center}
411
412 \begin{frame}
413
414 \begin{center}
415 \begin{tikzpicture}
416 \draw[>]
417
418 \end{tikzpicture}
419 \end{center}
420
421 \begin{frame}
422
423 \begin{center}
424 \begin{tikzpicture}
425 \draw[>]
426
427 \end{tikzpicture}
428 \end{center}
429
430 \begin{frame}
431
432 \begin{center}
433 \begin{tikzpicture}
434 \draw[>]
435
436 \end{tikzpicture}
437 \end{center}
438
439 \begin{frame}
440
441 \begin{center}
442 \begin{tikzpicture}
443 \draw[>]
444
445 \end{tikzpicture}
446 \end{center}
447
448 \begin{frame}
449
450 \begin{center}
451 \begin{tikzpicture}
452 \draw[>]
453
454 \end{tikzpicture}
455 \end{center}
456
457 \begin{frame}
458
459 \begin{center}
460 \begin{tikzpicture}
461 \draw[>]
462
463 \end{tikzpicture}
464 \end{center}
465
466 \begin{frame}
467
468 \begin{center}
469 \begin{tikzpicture}
470 \draw[>]
471
472 \end{tikzpicture}
473 \end{center}
474
475 \begin{frame}
476
477 \begin{center}
478 \begin{tikzpicture}
479 \draw[>]
480
481 \end{tikzpicture}
482 \end{center}
483
484 \begin{frame}
485
486 \begin{center}
487 \begin{tikzpicture}
488 \draw[>]
489
490 \end{tikzpicture}
491 \end{center}
492
493 \begin{frame}
494
495 \begin{center}
496 \begin{tikzpicture}
497 \draw[>]
498
499 \end{tikzpicture}
500 \end{center}
501
502 \begin{frame}
503
504 \begin{center}
505 \begin{tikzpicture}
506 \draw[>]
507
508 \end{tikzpicture}
509 \end{center}
510
511 \begin{frame}
512
513 \begin{center}
514 \begin{tikzpicture}
515 \draw[>]
516
517 \end{tikzpicture}
518 \end{center}
519
52
```

## Characteristics

- Command line or GUI
- Modal (one mode per class of action : edition, displacement, selection, etc...)
- Based on shortcuts and commands
- Fully customizable through plugin 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>

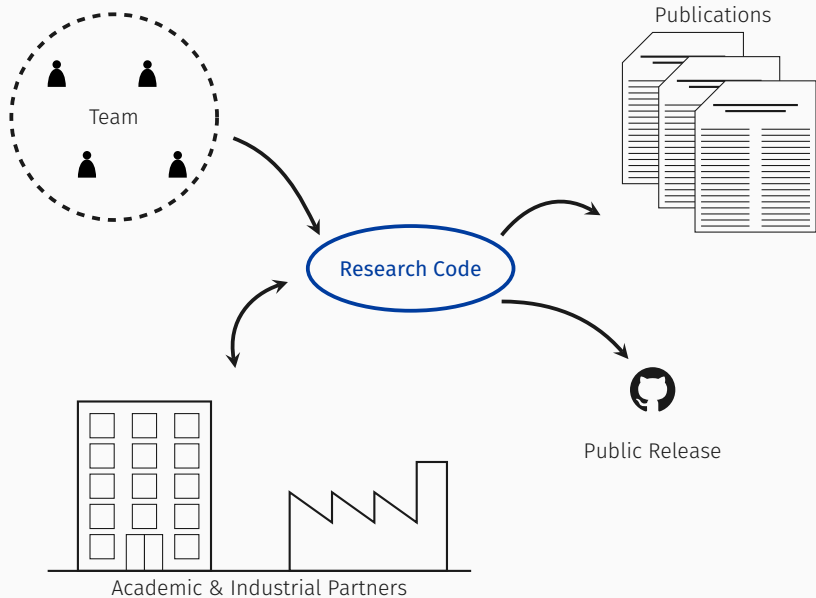
TODO

TODO

## Versionning for fun and profit

---

# Contribution to a research code



We need to **collaborate on** files every single day.

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



We need to **run tests on** files every single day.

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

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

Versionning is the way to go !

## Versionning for fun and profit

---

The old old way : names, zip, mail

# Versioning : the old old way

The first versioning attempt relies on 3 pillars :


- Naming conventions
- Regular backups
- Careful data management

But doesn't behave well when scaling...



- Team work complicated
- Modification conflicts
- No continuous time travel
- *etc.*

Some are even inventive and use different mail accounts to backup projects...

## Versionning : the old old way




 Awesome\_project

## Versionning : the old old way





-  Awesome\_project
-  Awesome\_project\_2








## Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup







## Versionning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test








## Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test









# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2










# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday










# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday
-  Awesome\_project\_publi

# Versioning : the old old way










-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday
-  Awesome\_project\_publi
-  Awesome\_project\_publi\_fix

# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday
-  Awesome\_project\_publi
-  Awesome\_project\_publi\_fix
-  Awesome\_project\_not\_good












# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday
-  Awesome\_project\_publi
-  Awesome\_project\_publi\_fix
-  Awesome\_project\_not\_good

...

# Versioning : the old old way

-  Awesome\_project
-  Awesome\_project\_2
-  Awesome\_project\_backup
-  Awesome\_project\_test
-  Awesome\_project\_test
-  Awesome\_project\_backup\_2
-  Awesome\_project\_idea\_sunday
-  Awesome\_project\_publi
-  Awesome\_project\_publi\_fix
-  Awesome\_project\_not\_good

...

# Versionning for fun and profit

---

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

## Versioning : the new old way

One more step towards real-time collaboration... using Dropbox & others...

- Free & paid plans
- Auto-sync even for big files
- Team notifications
- Mobile access to files

Possibility to track changes if a change-log is carefully maintained.

# Versioning : the new old way

One more step towards real-time collaboration... using Dropbox & others...

- Free & paid plans
- Auto-sync even for big files
- Team notifications
- Mobile access to files

Possibility to track changes if a change-log is carefully maintained.

But... is it really compliant with research prerequisites ?

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

— DropBox TOS

*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

*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



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

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

# Versionning for fun and profit

---

The badass way : **git**

# Versioning : the badass way

**git** was built to handle huge project (Linux Vanilla 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 (this version is working !)
- potentially server-less & self-hostable
- simultaneous modifications & smart merging
- usable through GUI (tig, gitk, tortoise-git & others) or command-line

# Versioning : the badass way

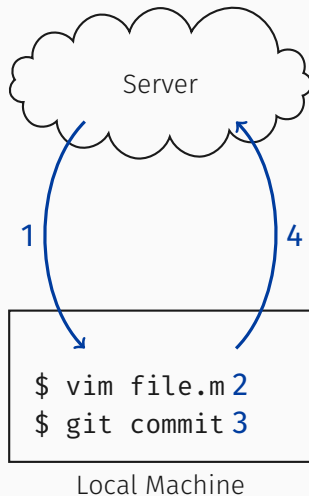
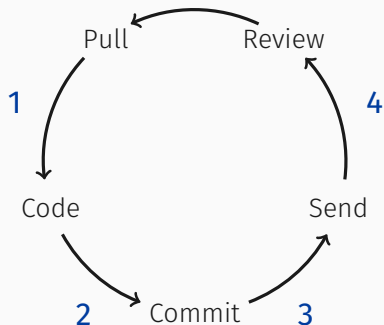
Concerning commits...

18ab98fc...	Unique (SHA1) identifier
Deletes Harkonnens	Changes Summary
Leto Atreides <leto@atreides.ar>	Committer & Author
Dec. 25 2015 21:42:21	Date & Time
- - + - - + +	Per-file additions/deletions
file doc.tex new file mode 10644 file stuff.sh old mode 100644 new mode 100755	Mode changes & file additions/deletions

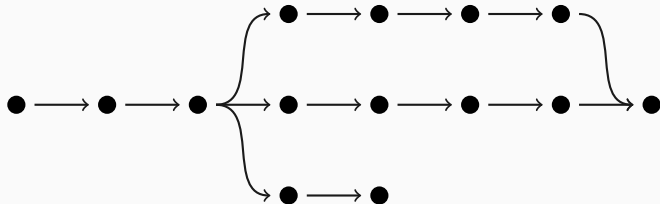
All the needed information is here !

# Versioning : the badass way

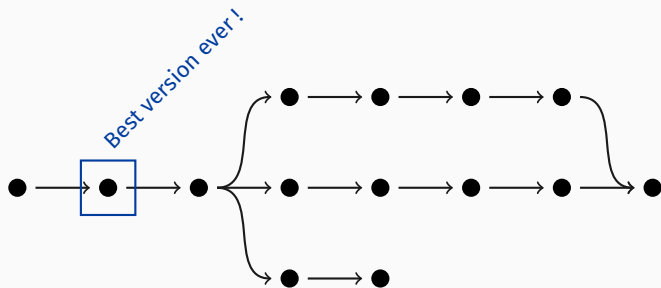
Collaboration-centric workflow



## Versioning : the badass way

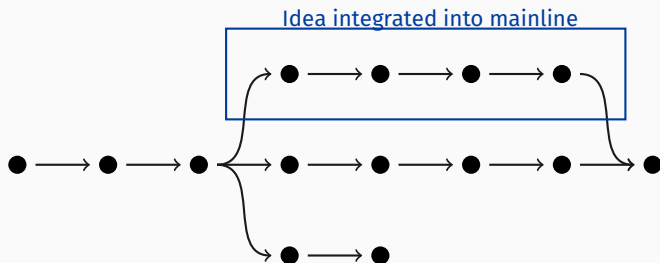


## Versioning : the badass way

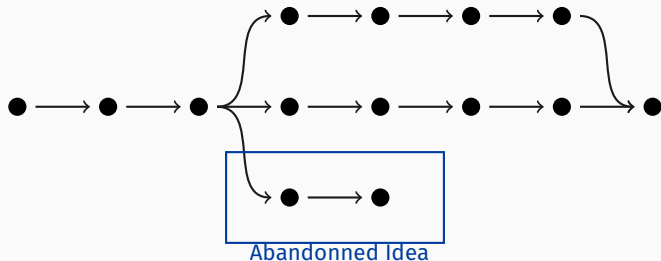




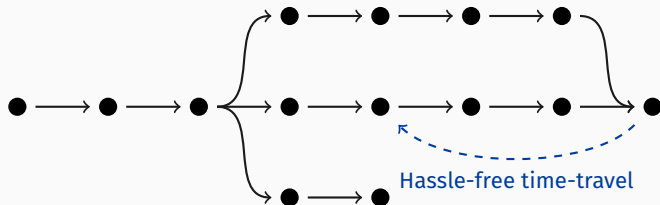
## Versioning : the badass way



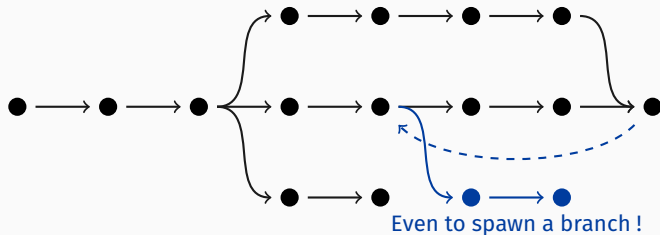
## Versioning : the badass way



## Versioning : the badass way



## Versioning : the badass way



Documentation is not a myth

---

Write The Fucking Manual

Write The ~~Fucking~~ Full Manual

## Write The ~~Fucking~~ Full Manual

Reasons ? Really ?

- Be able to reuse the code later (even after years)
- Be able to share without a full training for the receiver
- Allow reviewing & understanding the code
- Publish the code (y'know for reproducibility...)



## Documentation : right in the code

Nobody has **time** to write documentation...

but everybody can include comments in the code !

# Documentation : right in the code

Nobody has **time** to write documentation...

but everybody can include comments in the code !

## Different comments for different things

**File** What it does ? Bib. references ?

**Function** How is the task done ? How to use it ?

**Lines** Why is this operation done this way ?

# Documentation : right in the code

Nobody has **time** to write documentation...

but everybody can include comments in the code !

## Different comments for different things

**File** What it does ? Bib. references ?

**Function** How is the task done ? How to use it ?

**Lines** Why is this operation done this way ?

## Pro-tip

Start every single code file with a comment

*A computer program is just another way to express an idea.*

... and you have plenty of space to do so !

## Documentation : Variables naming matters

*A computer program is just another way to express an idea.*

... and you have plenty of space to do so !

Choose clear & meaningful names for variables and functions (avoid `tmp`, `data`, `this`, `process_data()`, `a-z` (except in loops), `id`, *etc.*)

*A computer program is just another way to express an idea.*

... and you have plenty of space to do so !

Choose clear & meaningful names for variables and functions (avoid `tmp`, `data`, `this`, `process_data()`, `a-z` (except in loops), `id`, etc.)

## Conventions

Formalized naming conventions exists (and some also address code styling issues, see PEP8 for example).

Use them as much as you can !

*When a task is repetitive... build a machine to do it.*

Documenting is boring but it's always the same !

Tools exist to automate doc generation :

**Doxygen** Doc-generator for C/C++/Java & others

**Sphinx** Doc-oriented static site generator build for the Python project (docutils) with EPUB, PDF,  $\text{\LaTeX}$  capabilities

**publish()** HTML/PDF publishing tool included in MATLAB

*etc.*

It's easy to publish it too : see [readthedocs.org](http://readthedocs.org)

What's next ?

---



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

Thank you !

Questions ?

manzari@kth.se – gaborit@kth.se