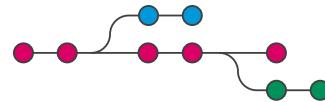


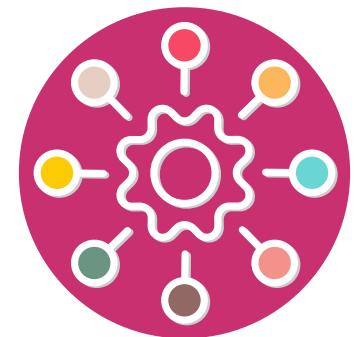


# Conception des systèmes Version Control



Filière Systèmes industriels

Silvan Zahno [silvan.zahno@hevs.ch](mailto:silvan.zahno@hevs.ch)





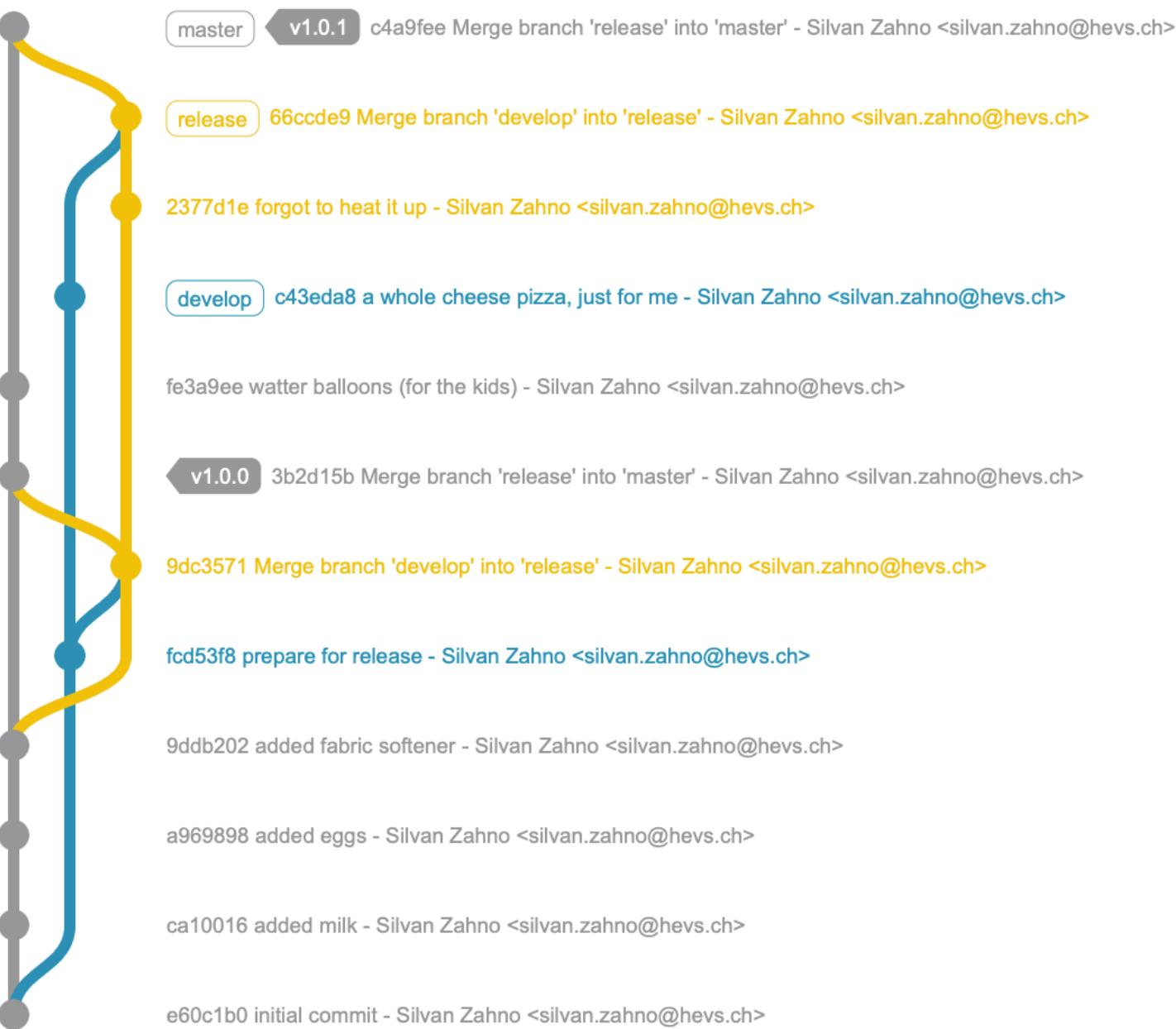
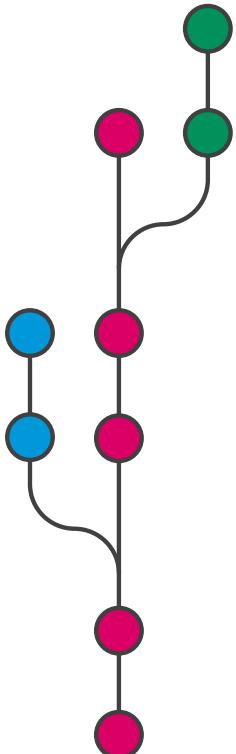
# Votre système actuel

```
.
```

- └── fichier important copy.txt
- └── fichier important v1.txt
- └── fichier important v2.1 backup.txt
- └── fichier important v2.1 backup.txt.old
- └── fichier important v2.1.txt
- └── fichier important v2.txt
- └── fichier important.txt

1 répertoire, 7 fichiers

# Version control





# Outils possibles

Git (git)



Subversion (svn)



Mercurial (hg)



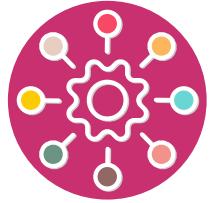
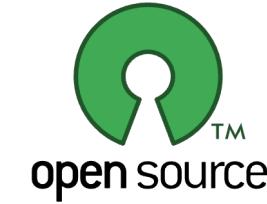
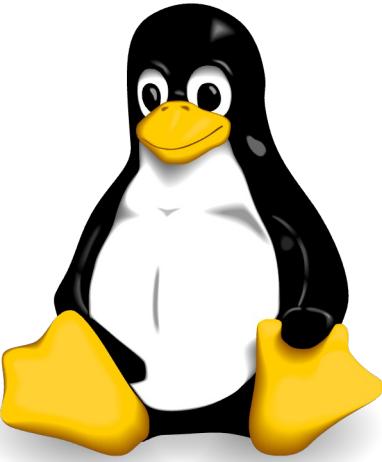
mercurial

SyD Version Control

Linux Torvalds

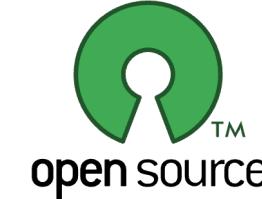
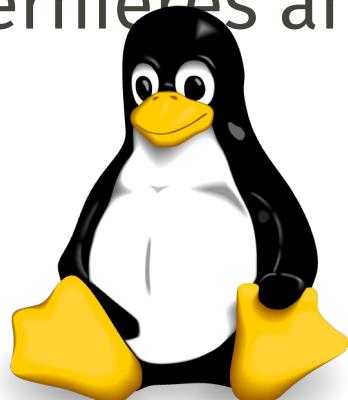
Linux (1991)

Git (2005)



# Pourquoi Linux a besoin de git

Linux est devenu le plus grand projet de développement collaboratif de l'histoire de l'informatique au cours des 30 dernières années.



- 600 distributions Linux actives
- 85% de tous les smartphones
- 500 superordinateurs de pointe
- > 27,8 millions de lignes de code
- > 12 000 contributeurs
- > 1 million de commits

<https://truelist.co/blog/linux-statistics/>

# Pourquoi pour un Power&Control ou Design&Material git est nécessaire



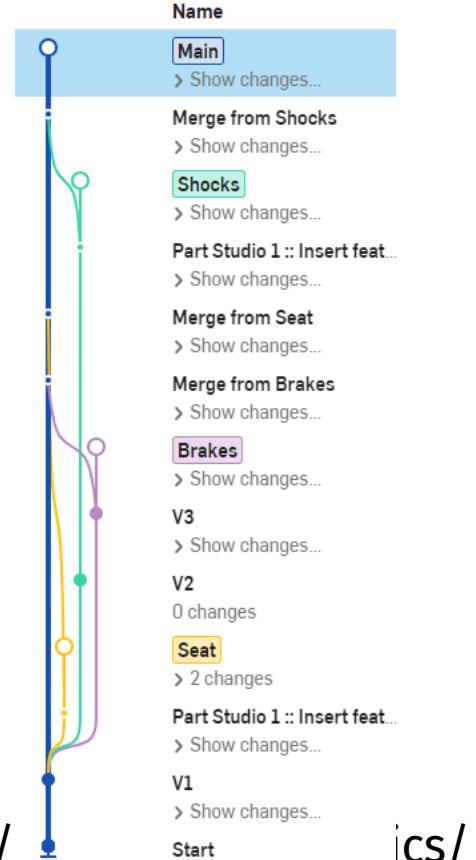
## Version

- Suivi des modifications
- Collaboration
- Capacités de retour en arrière
- Documentation
- Déploiement



**AUTODESK**  
Vault

<https://truelist.co/>





# Plateformes Git

## Gitlab

<https://gitlab.com>

<https://gitlab.hevs.ch>

## Github

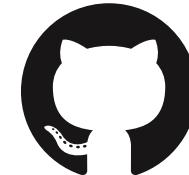
<https://github.com>

## Bitbucket

<https://bitbucket.com>



GitLab



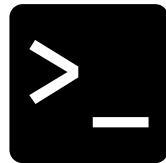
GitHub



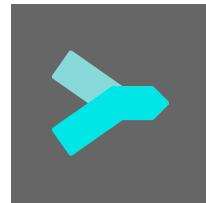
Bitbucket

# Outils Git

Ligne de commande



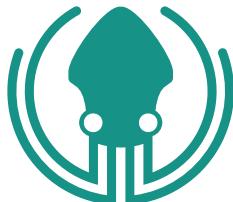
Sublime Merge



Git Cola



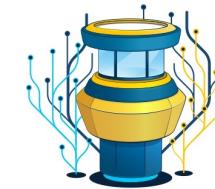
Git Kraken



Fork



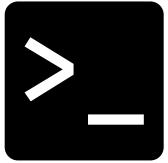
Tower



SmartGit



# Git ligne de commande



git status

git diff

git add <file>

git diff --staged

git reset <file>

git commit -m "<commit message>"

git fetch <remote>

git merge <remote> <branch>

git push <remote> <branch>

git pull

```
zas@zac:~$ git --help
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [<--exec-path=<path>] [<--html-path>] [<--man-path>] [<--info-path>
           [-p | --paginate | -P | --no-pager] [<--no-replace-objects>] [<--bare>
           [<--git-dir=<path>] [<--work-tree=<path>] [<--namespace=<name>]
           [<--super-prefix=<path>] [<--config-env=<name>>=<envvar>]
           [<command> [<args>]]
```

These are common Git commands used in various situations:

```
start a working area (see also: git help tutorial)
clone    Clone a repository into a new directory
init     Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
add      Add file contents to the index
mv      Move or rename a file, a directory, or a symlink
restore  Restore working tree files
rm      Remove files from the working tree and from the index

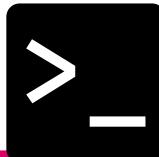
examine the history and state (see also: git help revisions)
bisect   Use binary search to find the commit that introduced a bug
diff     Show changes between commits, commit and working tree, etc
grep     Print lines matching a pattern
log      Show commit logs
show    Show various types of objects
status   Show the working tree status

grow, mark and tweak your common history
branch   List, create, or delete branches
commit   Record changes to the repository
merge    Join two or more development histories together
rebase   Reapply commits on top of another base tip
reset   Reset current HEAD to the specified state
switch  Switch branches
tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
fetch   Download objects and refs from another repository
pull    Fetch from and integrate with another repository or a local branch
push    Update remote refs along with associated objects

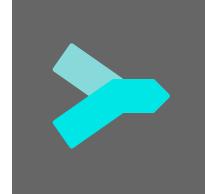
'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.
```

# git commands



Command	Description	Command	Description
<b>Démarrer une zone de travail</b>		<b>Développez, marquez et modifiez votre histoire commune</b>	
clone	Cloner un dépôt dans un nouveau répertoire	branch	Lister, créer ou supprimer des branches
init	Créer un dépôt Git vide ou réinitialiser un dépôt existant	checkout	Changer de branche ou restaurer les fichiers de l'arborescence de travail
<b>Travailler sur la modification en cours</b>		commit	Enregistrer les modifications apportées à la base de données
add	Ajouter le contenu d'un fichier à l'index	diff	Afficher les changements entre les livraisons, les livraisons et l'arbre de travail, etc.
mv	Déplacer ou renommer un fichier, un répertoire ou un lien symbolique	merge	Joindre deux ou plusieurs historiques de développement
reset	Réinitialiser le HEAD actuel à l'état spécifié	rebase	Réappliquer les commits par-dessus un autre conseil de base
rm	Supprimer des fichiers de l'arbre de travail et de l'index	tag	Créer, lister, supprimer ou vérifier un objet d'étiquette
<b>Examiner l'historique et l'état</b>		<b>Collaborer</b>	
log	Afficher les journaux de livraison	fetch	Télécharger des objets et des références à partir d'un autre dépôt
show	Afficher différents types d'objets	pull	Récupérer et intégrer des objets d'une autre base de données ou d'une branche locale
status	Afficher l'état de l'arbre de travail	push	Mettre à jour les références distantes ainsi que les objets associés

# Sublime Merge



~/work/repo/edu/car/car-course

LICENSE UPGRADE REQUIRED

master

Commits Files Summary

LOCATIONS

BRANCHES (1) master

REMOTES (1) origin

TAGS (0) master

STASHES (0)

SUBMODULES (0)

1 unstaged file Commit Changes

Merge remote-tracking branch 'origin/master' 16

CHG: updated planning zas master origin/master Thu 08:11

ADD: ALU and ImmSrc doc Axam Thu, 13 Apr 15:19

ADD: EBS2/EBS3 specs Axam Tue, 11 Apr 15:25

FIX: memory stack images zas Tue, 4 Apr 11:00

FIX: errors in immediate and type images zas Tue, 4 Apr 07:46

ADD: files in arc exercises zas Mon, 3 Apr 13:30

ADD: note on Ripes memory management Axam Fri, 31 Mar 15:18

CHG: Planning zas Fri, 31 Mar 08:45

FIX: reverse engineering solution Axam Thu, 30 Mar 17:25

FIX: ISA syntax errors zas Tue, 28 Mar 07:59

Merge remote-tracking branch 'origin/master' 6

zас Tue, 28 Mar 07:29

FIX: errors in ISA zас Tue, 28 Mar 07:29

ADD: windows Geekbench window Axam Thu, 16 Mar 10:14

FIX: add scripts folder Axel Amand Thu, 16 Mar 09:31

REM: car-heirv and car-labs doc deployment Axel Amand Thu, 16 Mar 09:18

CHG: BEM labo from geekbench 5 to 6 zас Tue, 14 Mar 14:25

UPD: all PDFs Axam Wed, 8 Mar 19:10

FIX: errors in ARC, ISA, FUN and PER slides zас Tue, 7 Mar 09:09

Commit Hash 702c8ff178adbf6639884c48b72e4bc68361d13c f163cea8c5f992b7c78549993694b6670e0da8b

Tree zas<silvan.zahn@nev.ch>

Author zas<silvan.zahn@nev.ch>

Date Thu, 20 Apr 2023 08:12

Parents 6e927ef6, 68810079

Branches master origin/master

Stats 16 files changed: 15 +10

Merge remote-tracking branch 'origin/master'

Collapse all

labo/latex/b2-content/scr/00-solution.tex -1 +5

Subsection: Simulation:

```
164 done;
165    beq x2, x2, main      # infinite loop
166  end(minted)
167 \newline\nullnewline
168 Each instruction takes one clock cycle => 19 are executed (addi x5, x0, 0 not
executed because of previous beq; addi x2, x0, 1 not executed because of jal).
=> 19/6M = 287.9 ns.
```

Subsection: Simulation:

```
164 done;
165    beq x2, x2, main      # infinite loop
166  end(minted)
167 \newline\nullnewline
168 Each instruction takes one clock cycle => 19 are executed (addi x5, x0, 0 not
executed because of previous beq; addi x2, x0, 1 not executed because of jal).
=> 19/6M = 287.9 ns.
```

On the EBS2 board @ 66MHz \rightarrow \\$T\_{exec} = \frac{nb\\_cycles}{F\_{sys}} = \frac{19}{(19)(6M)} = 287.9 ns.

On the EBS3 board @ 50MHz \rightarrow \\$T\_{exec} = \frac{nb\\_cycles}{F\_{sys}} = \frac{19}{(19)(5M)} = 380 ns.

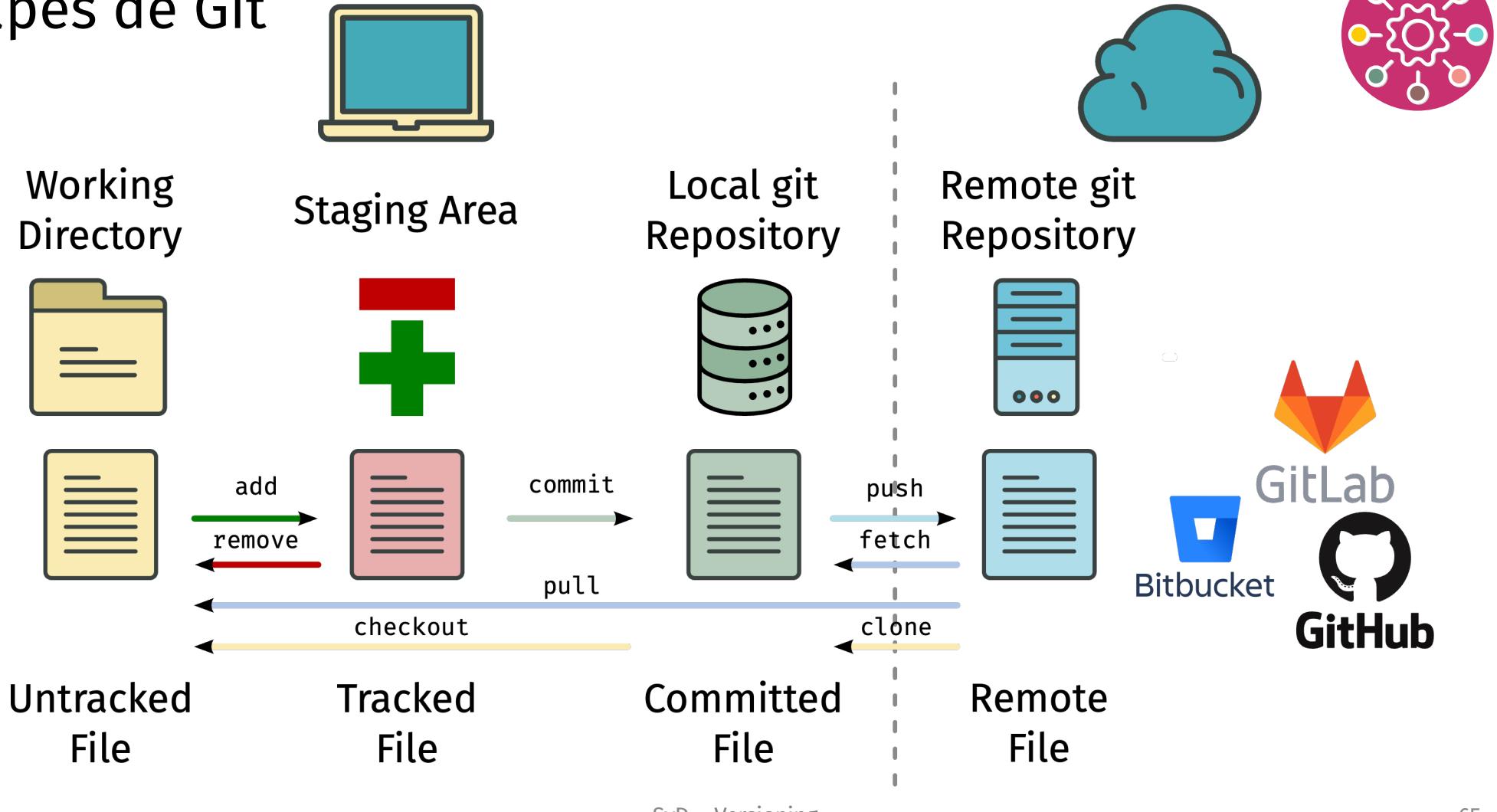
Subsection: Umsetzung:

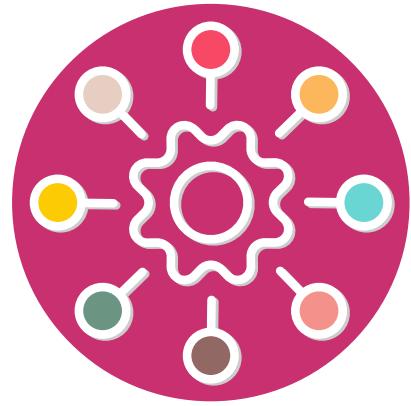
```
164 Le \textbf{mainDecoder} peut être écrit en VHDL. Pour cela, vous pouvez analyser
l'exemple de code \ref{fig:riscv-mainDecoder-code} ci-dessous et l'adapter en
conséquence.
165
166 \textbf{Note:} Dans HDL Designer, lorsque vous sélectionnez le type de contenu d'un
bloc, choisissez \textbf{VHDL File -> Architecture}, et contrôlez que le
langage soit défini sur \textbf{VHDL 2008}. Sur la page suivante, \textbf{Architecture}
correspond au nom de la vue (un bloc peut avoir différents
contenus) et \textbf{Entity} au nom du bloc (\textbf{mainDecoder} par exemple).}
167 }
168 \opt{d}{%
169 Schreiben Sie hierzu für beide Subblöcke, \textbf{mainDecoder} sowie \textbf{ALUDecoder}, eine Wahrheitstabelle für alle benötigten Instruktionen.
170 \opt{f}{\caption{figure}{Exemple de code MainDecoder}}
171 \opt{d}{\caption{figure}{MainDecoder Code-Beispiel}}
172 \label{fig:riscv-mainDecoder-code}
173
174 \subsubsection{ALU}
175 \opt{f}{%
176 L'ALU réalise les fonctions arithmétiques et logiques selon la table suivante:
177 }
178 \opt{d}{%
179 Die ALU realisiert die arithmetischen und logischen Funktionen gemäß der folgenden
Tabelle:
180 }
181
182 \begin{table}[h]
```

g:riscv-mainDecoder-code:

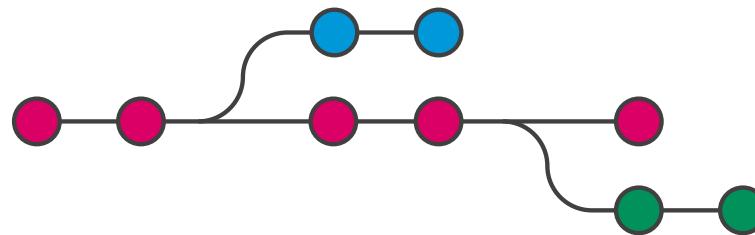
```
110 \opt{f}{\caption{figure}{Exemple de code MainDecoder}}
111 \opt{d}{\caption{figure}{MainDecoder Code-Beispiel}}
112 \label{fig:riscv-mainDecoder-code}
113
114 \subsubsection{ALU}
115 \opt{f}{%
116 L'ALU réalise les fonctions arithmétiques et logiques selon la table suivante:
117 }
118 \opt{d}{%
119 Die ALU realisiert die arithmetischen und logischen Funktionen gemäß der folgenden
Tabelle:
120 }
121
122 \begin{table}[h]
```

# Étapes de Git





## Git Branch et Merge Exemple

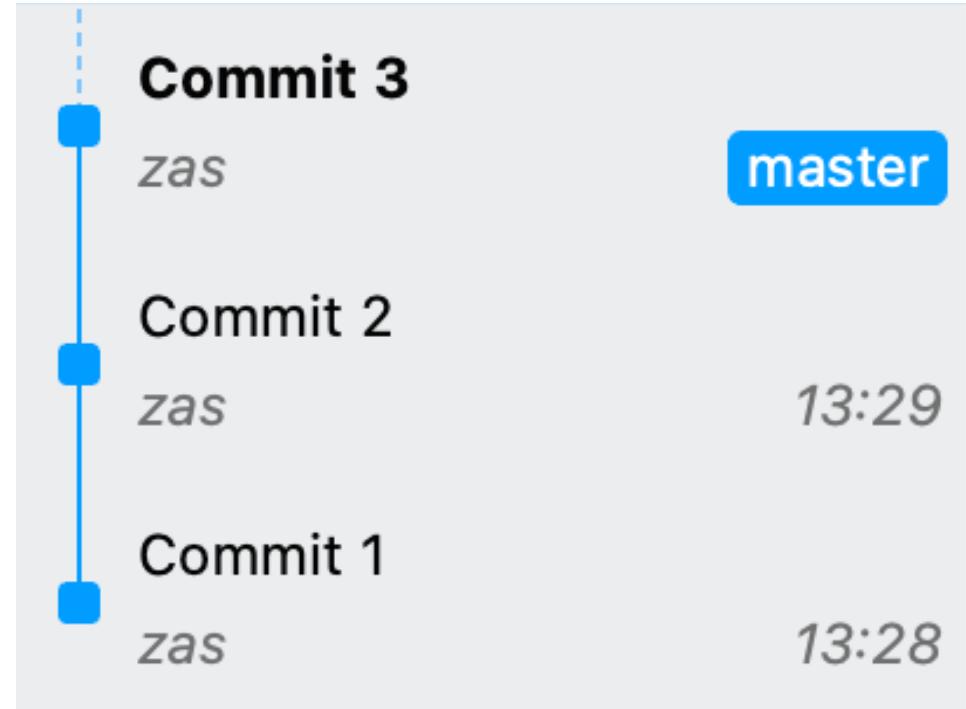




# Branch et Merge

## Initial repo state

Chaque repo a une branche **main** ou une branche **master** comme branche par défaut.



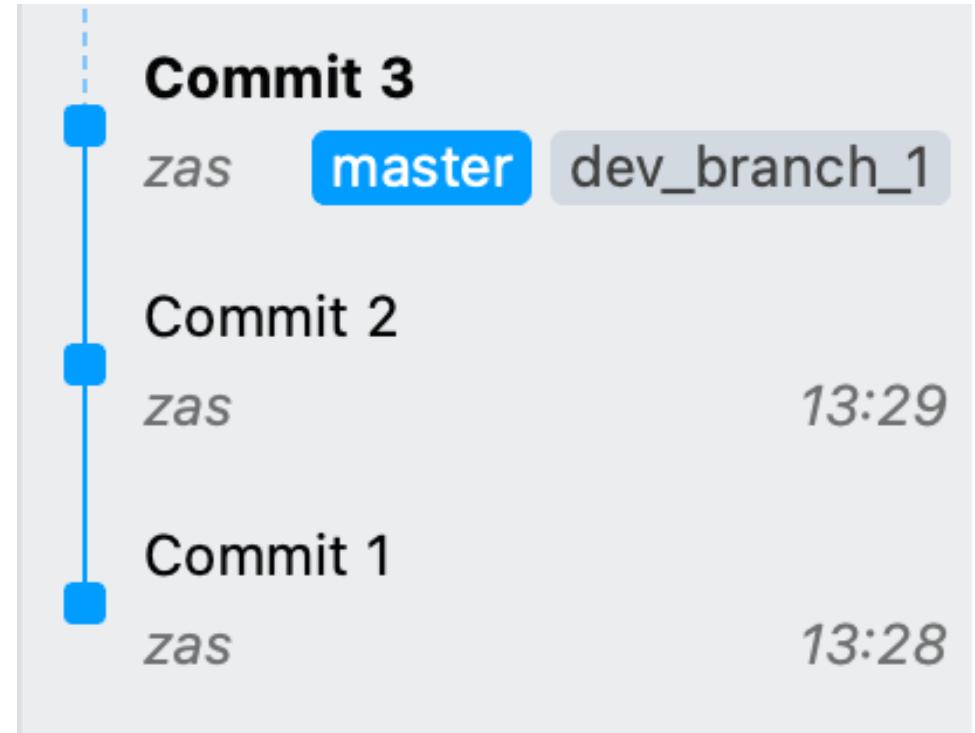


# Branch et Merge

Create branch dev\_branch\_1

Create branch dev\_branch\_1

```
$ git branch dev_branch_1
```





## Branch et Merge

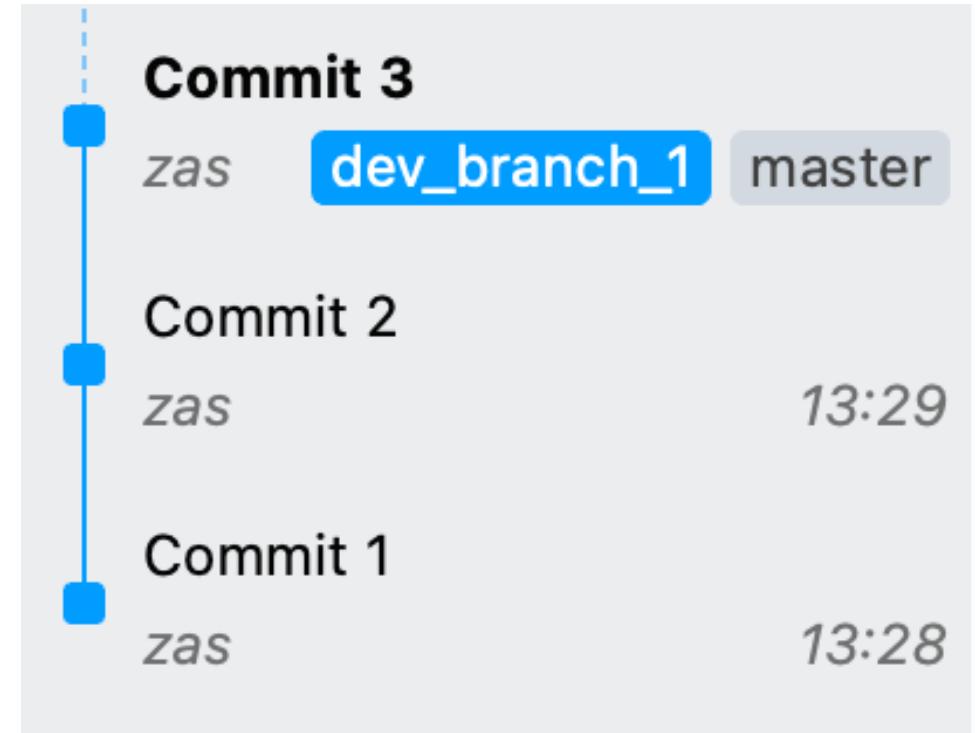
**Checkout** branch dev\_branch\_1

Vérifier sur quelle branche nous nous trouvons

```
$ git branch
```

Checkout branch dev\_branch\_1

```
$ git checkout dev_branch_1
```





# Branch et Merge

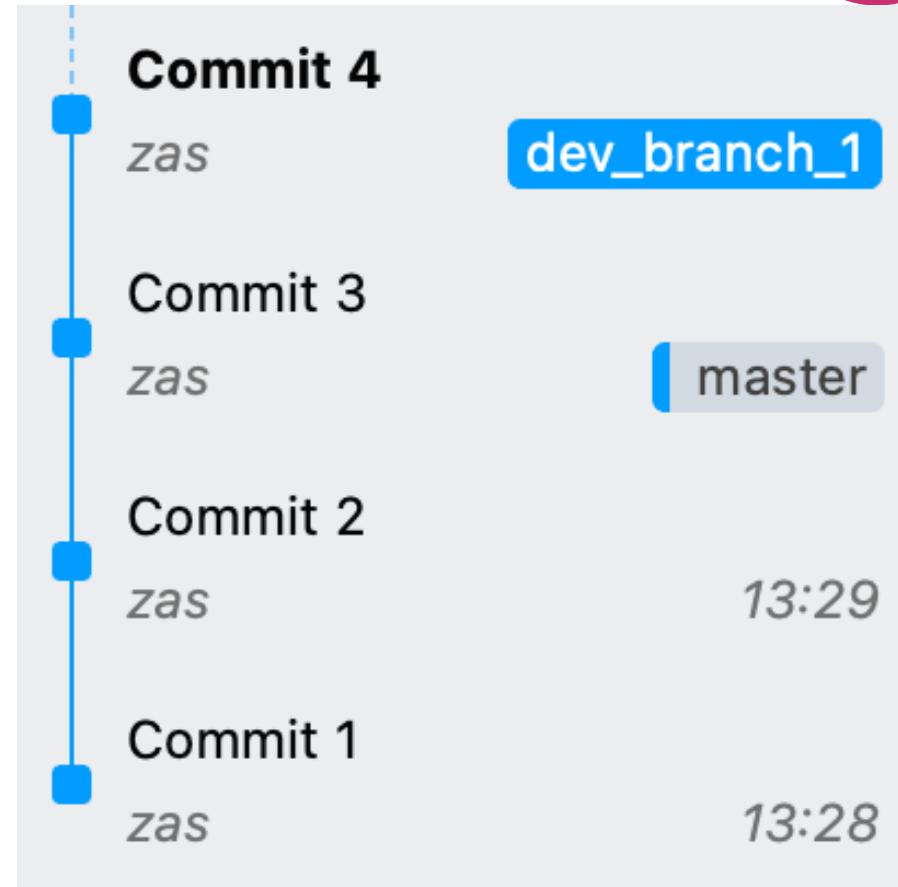
**Commit** on dev\_branch\_1

Stage new file

```
$ git add file.md
```

Commit stages files

```
$ git commit -m "Commit 4"
```





# Branch et Merge

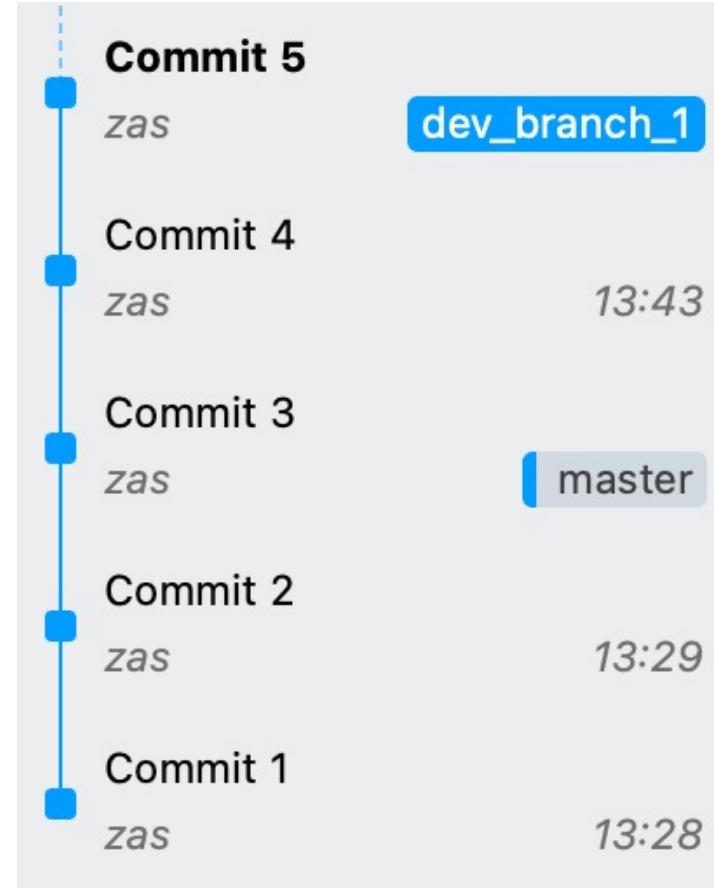
**Commit** on dev\_branch\_1

Stage new file

```
$ git add file.md
```

Commit stages files

```
$ git commit -m "Commit 5"
```





# Branch et Merge

**Commit** on master branch

Checkout master branch

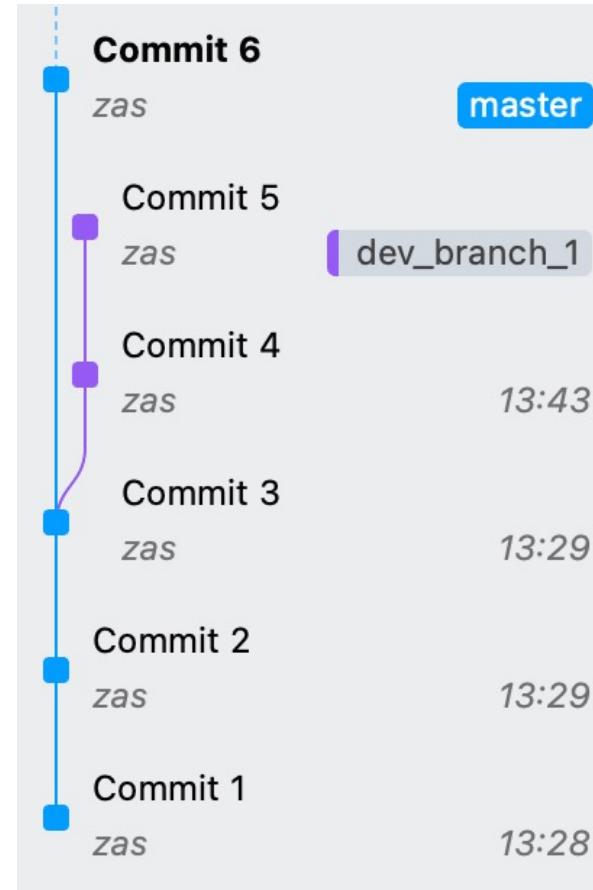
```
$ git checkout master
```

Stage new file

```
$ git add file.md
```

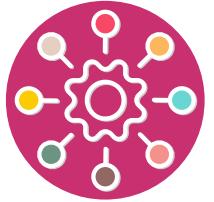
Commit stages files

```
$ git commit -m "Commit 6"
```



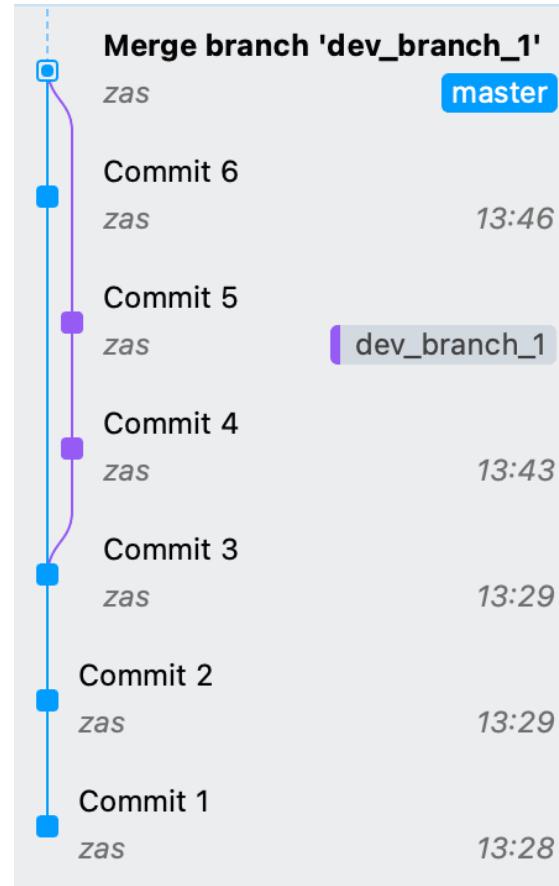
# Branch et Merge

Three way merge master and dev\_branch\_1

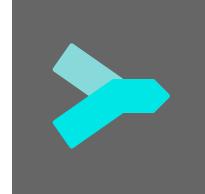


Merge dev\_branch\_1 into master

```
$ git merge dev_branch_1
```



# Demo



~/work/repo/edu/car/car-course

LICENSE UPGRADE REQUIRED

Locations

BRANCHES (1) master

REMOTES (1) origin

TAGS (0)

STASHES (0)

SUBMODULES (0)

Commits

Files

Summary

Commit Hash 702c8f17838ad6639884c48b72e4bc68361d13c  
Tree f163cea8c55f992b7c7854993694b6670e0da8b  
Author zas<silvan.zahn@nev.ch>  
Date Thu, 20 Apr 2023 08:12  
Parents 6e927ef6, 68810079  
Branches master origin/master  
Stats 16 files changed: 15 +10

Merge remote-tracking branch 'origin/master'

CHG: updated planning

ADD: ALU and ImmSrc doc

ADD: EBS2/EBS3 specs

FIX: memory stack images

FIX: errors in immediate and type images

ADD: files in arc exercises

ADD: note on Ripes memory management

CHG: Planning

FIX: reverse engineering solution

FIX: ISA syntax errors

Merge remote-tracking branch 'origin/master'

FIX: errors in ISA

ADD: windows Geekbench window

FIX: add scripts folder

REM: car-hevri and car-labs doc deployment

CHG: BEM labo from geekbench 5 to 6

UPD: all PDFs

FIX: errors in ARC, ISA, FUN and PER slides

master

Thu 08:11

Thu, 13 Apr 15:19

Tue, 11 Apr 15:25

Tue, 4 Apr 11:00

Tue, 4 Apr 07:46

Mon, 3 Apr 13:30

Fri, 31 Mar 15:18

Fri, 31 Mar 08:45

Thu, 30 Mar 17:25

Tue, 28 Mar 07:59

Thu, 16 Mar 10:14

Thu, 16 Mar 09:31

Thu, 16 Mar 09:18

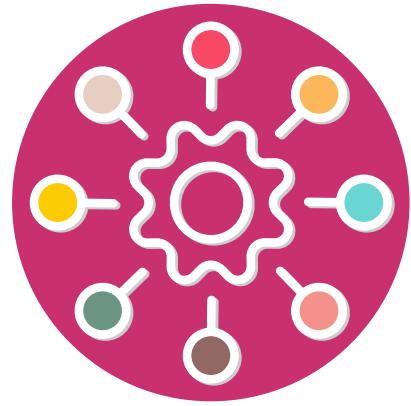
Tue, 14 Mar 14:25

Wed, 8 Mar 19:10

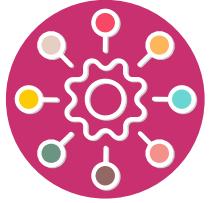
Tue, 7 Mar 09:09

00-solution.tex 03-controlunit.tex 04-simulation.tex 05-deployment.tex 05-deployment\_ebs3.tex 08-CAr-Labor-SCR-d.pdf 08-CAr-Labor-SCR-f.pdf CAr-Labor-SCR-s.pdf CAr-Labor-SCR-t.pdf

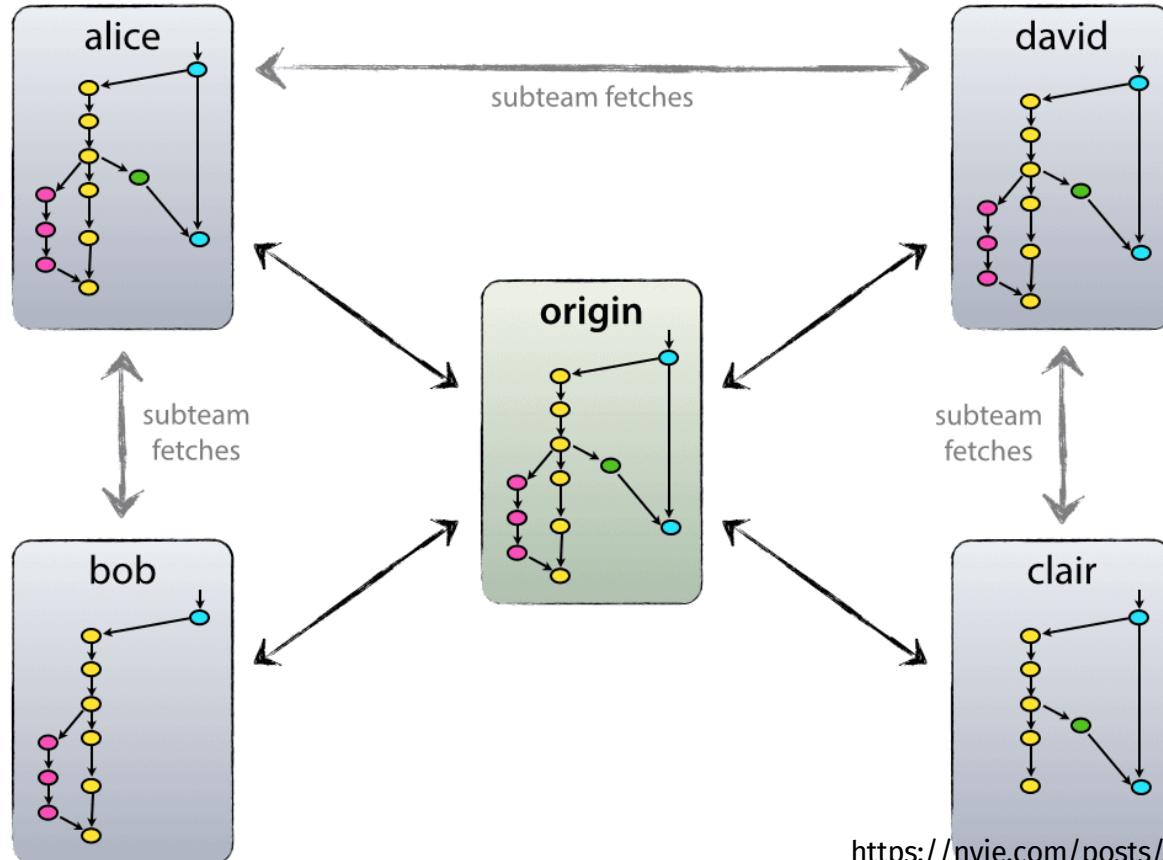
...  
Collapse all  
...  
Subsection: Simulation:  
done:  
bed x2, x2, main # infinite loop  
end(minted)  
164 done:  
165 bed x2, x2, main # infinite loop  
166 end(minted)  
167 \newline\nullnewline  
168 Each instruction takes one clock cycle => 19 are executed (addi x5, x0, 0 not executed because of previous beq; addi x2, x0, 1 not executed because of jal).  
169 On the EBS2 board @ 66MHz \rightarrow \\$T\_{exec} = \frac{nb\\_cycles}{F\_{sys}} = \frac{19}{(19)(66M)} = 287.9 ns.  
170 On the EBS3 board @ 50MHz \rightarrow \\$T\_{exec} = \frac{nb\\_cycles}{F\_{sys}} = \frac{19}{(19)(50M)} = 380 ns.  
171  
172  
173  
174  
175 ...  
Subsection: Umsetzung:  
Le \textbf{mainDecoder} peut être écrit en VHDL. Pour cela, vous pouvez analyser l'exemple de code \ref{fig:riscv-mainDecoder-code} ci-dessous et l'adapter en conséquence.  
InnoDB (Dans HDL Designer, lorsque vous sélectionnez le type de contenu d'un bloc, choisissez \textit{VHDL File -> Architecture}, et contrôlez que le langage soit défini sur \textit{VHDL 2008}). Sur la page suivante, \textit{Architecture} correspond au nom de la vue (un bloc peut avoir différents contenus) et \textit{Entity} au nom du bloc (\textit{mainDecoder} par exemple).  
Schreiben Sie hierzu für beide Subblöcke, \textit{mainDecoder} sowie \textit{ALUDecoder}, eine Wahrheitstabelle für alle benötigten Instruktionen.  
g:riscv-mainDecoder-code:  
110 \opt{f}{\caption{figure}{Exemple de code MainDecoder}}  
111 \opt{d}{\caption{figure}{MainDecoder Code-Beispiel}}  
112 \label{fig:riscv-mainDecoder-code}  
113  
114 \subsubsection{ALU}  
115 \opt{f}{  
116 L'ALU réalise les fonctions arithmétiques et logiques selon la table suivante:  
117 }  
118 \opt{d}{  
119 Die ALU realisiert die arithmetischen und logischen Funktionen gemäß der folgenden Tabelle:  
120 }  
121  
122 ...  
123 ...  
124 ...  
125 ...  
126 ...  
127 ...  
128 ...  
129 ...  
130 ...  
131 ...  
132 ...  
133 ...  
134 ...  
135 ...  
136 ...  
137 ...  
138 ...  
139 ...  
140 ...  
141 ...  
142 ...  
143 ...  
144 ...  
145 ...  
146 ...  
147 ...  
148 ...  
149 ...  
150 ...  
151 ...  
152 ...  
153 ...  
154 ...  
155 ...  
156 ...  
157 ...  
158 ...  
159 ...  
160 ...  
161 ...  
162 ...  
163 ...  
164 ...  
165 ...  
166 ...  
167 ...  
168 ...  
169 ...  
170 ...  
171 ...  
172 ...  
173 ...  
174 ...  
175 ...  
176 ...  
177 ...  
178 ...  
179 ...  
180 ...  
181 ...  
182 ...  
183 ...  
184 ...  
185 ...  
186 ...  
187 ...  
188 ...  
189 ...  
190 ...  
191 ...  
192 ...  
193 ...  
194 ...  
195 ...  
196 ...  
197 ...  
198 ...  
199 ...  
200 ...  
201 ...  
202 ...  
203 ...  
204 ...  
205 ...  
206 ...  
207 ...  
208 ...  
209 ...  
210 ...  
211 ...  
212 ...  
213 ...  
214 ...  
215 ...  
216 ...  
217 ...  
218 ...  
219 ...  
220 ...  
221 ...  
222 ...  
223 ...  
224 ...  
225 ...  
226 ...  
227 ...  
228 ...  
229 ...  
230 ...  
231 ...  
232 ...  
233 ...  
234 ...  
235 ...  
236 ...  
237 ...  
238 ...  
239 ...  
240 ...  
241 ...  
242 ...  
243 ...  
244 ...  
245 ...  
246 ...  
247 ...  
248 ...  
249 ...  
250 ...  
251 ...  
252 ...  
253 ...  
254 ...  
255 ...  
256 ...  
257 ...  
258 ...  
259 ...  
260 ...  
261 ...  
262 ...  
263 ...  
264 ...  
265 ...  
266 ...  
267 ...  
268 ...  
269 ...  
270 ...  
271 ...  
272 ...  
273 ...  
274 ...  
275 ...  
276 ...  
277 ...  
278 ...  
279 ...  
280 ...  
281 ...  
282 ...  
283 ...  
284 ...  
285 ...  
286 ...  
287 ...  
288 ...  
289 ...  
290 ...  
291 ...  
292 ...  
293 ...  
294 ...  
295 ...  
296 ...  
297 ...  
298 ...  
299 ...  
300 ...  
301 ...  
302 ...  
303 ...  
304 ...  
305 ...  
306 ...  
307 ...  
308 ...  
309 ...  
310 ...  
311 ...  
312 ...  
313 ...  
314 ...  
315 ...  
316 ...  
317 ...  
318 ...  
319 ...  
320 ...  
321 ...  
322 ...  
323 ...  
324 ...  
325 ...  
326 ...  
327 ...  
328 ...  
329 ...  
330 ...  
331 ...  
332 ...  
333 ...  
334 ...  
335 ...  
336 ...  
337 ...  
338 ...  
339 ...  
340 ...  
341 ...  
342 ...  
343 ...  
344 ...  
345 ...  
346 ...  
347 ...  
348 ...  
349 ...  
350 ...  
351 ...  
352 ...  
353 ...  
354 ...  
355 ...  
356 ...  
357 ...  
358 ...  
359 ...  
360 ...  
361 ...  
362 ...  
363 ...  
364 ...  
365 ...  
366 ...  
367 ...  
368 ...  
369 ...  
370 ...  
371 ...  
372 ...  
373 ...  
374 ...  
375 ...  
376 ...  
377 ...  
378 ...  
379 ...  
380 ...  
381 ...  
382 ...  
383 ...  
384 ...  
385 ...  
386 ...  
387 ...  
388 ...  
389 ...  
390 ...  
391 ...  
392 ...  
393 ...  
394 ...  
395 ...  
396 ...  
397 ...  
398 ...  
399 ...  
400 ...  
401 ...  
402 ...  
403 ...  
404 ...  
405 ...  
406 ...  
407 ...  
408 ...  
409 ...  
410 ...  
411 ...  
412 ...  
413 ...  
414 ...  
415 ...  
416 ...  
417 ...  
418 ...  
419 ...  
420 ...  
421 ...  
422 ...  
423 ...  
424 ...  
425 ...  
426 ...  
427 ...  
428 ...  
429 ...  
430 ...  
431 ...  
432 ...  
433 ...  
434 ...  
435 ...  
436 ...  
437 ...  
438 ...  
439 ...  
440 ...  
441 ...  
442 ...  
443 ...  
444 ...  
445 ...  
446 ...  
447 ...  
448 ...  
449 ...  
450 ...  
451 ...  
452 ...  
453 ...  
454 ...  
455 ...  
456 ...  
457 ...  
458 ...  
459 ...  
460 ...  
461 ...  
462 ...  
463 ...  
464 ...  
465 ...  
466 ...  
467 ...  
468 ...  
469 ...  
470 ...  
471 ...  
472 ...  
473 ...  
474 ...  
475 ...  
476 ...  
477 ...  
478 ...  
479 ...  
480 ...  
481 ...  
482 ...  
483 ...  
484 ...  
485 ...  
486 ...  
487 ...  
488 ...  
489 ...  
490 ...  
491 ...  
492 ...  
493 ...  
494 ...  
495 ...  
496 ...  
497 ...  
498 ...  
499 ...  
500 ...  
501 ...  
502 ...  
503 ...  
504 ...  
505 ...  
506 ...  
507 ...  
508 ...  
509 ...  
510 ...  
511 ...  
512 ...  
513 ...  
514 ...  
515 ...  
516 ...  
517 ...  
518 ...  
519 ...  
520 ...  
521 ...  
522 ...  
523 ...  
524 ...  
525 ...  
526 ...  
527 ...  
528 ...  
529 ...  
530 ...  
531 ...  
532 ...  
533 ...  
534 ...  
535 ...  
536 ...  
537 ...  
538 ...  
539 ...  
540 ...  
541 ...  
542 ...  
543 ...  
544 ...  
545 ...  
546 ...  
547 ...  
548 ...  
549 ...  
550 ...  
551 ...  
552 ...  
553 ...  
554 ...  
555 ...  
556 ...  
557 ...  
558 ...  
559 ...  
560 ...  
561 ...  
562 ...  
563 ...  
564 ...  
565 ...  
566 ...  
567 ...  
568 ...  
569 ...  
570 ...  
571 ...  
572 ...  
573 ...  
574 ...  
575 ...  
576 ...  
577 ...  
578 ...  
579 ...  
580 ...  
581 ...  
582 ...  
583 ...  
584 ...  
585 ...  
586 ...  
587 ...  
588 ...  
589 ...  
590 ...  
591 ...  
592 ...  
593 ...  
594 ...  
595 ...  
596 ...  
597 ...  
598 ...  
599 ...  
600 ...  
601 ...  
602 ...  
603 ...  
604 ...  
605 ...  
606 ...  
607 ...  
608 ...  
609 ...  
610 ...  
611 ...  
612 ...  
613 ...  
614 ...  
615 ...  
616 ...  
617 ...  
618 ...  
619 ...  
620 ...  
621 ...  
622 ...  
623 ...  
624 ...  
625 ...  
626 ...  
627 ...  
628 ...  
629 ...  
630 ...  
631 ...  
632 ...  
633 ...  
634 ...  
635 ...  
636 ...  
637 ...  
638 ...  
639 ...  
640 ...  
641 ...  
642 ...  
643 ...  
644 ...  
645 ...  
646 ...  
647 ...  
648 ...  
649 ...  
650 ...  
651 ...  
652 ...  
653 ...  
654 ...  
655 ...  
656 ...  
657 ...  
658 ...  
659 ...  
660 ...  
661 ...  
662 ...  
663 ...  
664 ...  
665 ...  
666 ...  
667 ...  
668 ...  
669 ...  
670 ...  
671 ...  
672 ...  
673 ...  
674 ...  
675 ...  
676 ...  
677 ...  
678 ...  
679 ...  
680 ...  
681 ...  
682 ...  
683 ...  
684 ...  
685 ...  
686 ...  
687 ...  
688 ...  
689 ...  
690 ...  
691 ...  
692 ...  
693 ...  
694 ...  
695 ...  
696 ...  
697 ...  
698 ...  
699 ...  
700 ...  
701 ...  
702 ...  
703 ...  
704 ...  
705 ...  
706 ...  
707 ...  
708 ...  
709 ...  
710 ...  
711 ...  
712 ...  
713 ...  
714 ...  
715 ...  
716 ...  
717 ...  
718 ...  
719 ...  
720 ...  
721 ...  
722 ...  
723 ...  
724 ...  
725 ...  
726 ...  
727 ...  
728 ...  
729 ...  
730 ...  
731 ...  
732 ...  
733 ...  
734 ...  
735 ...  
736 ...  
737 ...  
738 ...  
739 ...  
740 ...  
741 ...  
742 ...  
743 ...  
744 ...  
745 ...  
746 ...  
747 ...  
748 ...  
749 ...  
750 ...  
751 ...  
752 ...  
753 ...  
754 ...  
755 ...  
756 ...  
757 ...  
758 ...  
759 ...  
760 ...  
761 ...  
762 ...  
763 ...  
764 ...  
765 ...  
766 ...  
767 ...  
768 ...  
769 ...  
770 ...  
771 ...  
772 ...  
773 ...  
774 ...  
775 ...  
776 ...  
777 ...  
778 ...  
779 ...  
780 ...  
781 ...  
782 ...  
783 ...  
784 ...  
785 ...  
786 ...  
787 ...  
788 ...  
789 ...  
790 ...  
791 ...  
792 ...  
793 ...  
794 ...  
795 ...  
796 ...  
797 ...  
798 ...  
799 ...  
800 ...  
801 ...  
802 ...  
803 ...  
804 ...  
805 ...  
806 ...  
807 ...  
808 ...  
809 ...  
810 ...  
811 ...  
812 ...  
813 ...  
814 ...  
815 ...  
816 ...  
817 ...  
818 ...  
819 ...  
820 ...  
821 ...  
822 ...  
823 ...  
824 ...  
825 ...  
826 ...  
827 ...  
828 ...  
829 ...  
830 ...  
831 ...  
832 ...  
833 ...  
834 ...  
835 ...  
836 ...  
837 ...  
838 ...  
839 ...  
840 ...  
841 ...  
842 ...  
843 ...  
844 ...  
845 ...  
846 ...  
847 ...  
848 ...  
849 ...  
850 ...  
851 ...  
852 ...  
853 ...  
854 ...  
855 ...  
856 ...  
857 ...  
858 ...  
859 ...  
860 ...  
861 ...  
862 ...  
863 ...  
864 ...  
865 ...  
866 ...  
867 ...  
868 ...  
869 ...  
870 ...  
871 ...  
872 ...  
873 ...  
874 ...  
875 ...  
876 ...  
877 ...  
878 ...  
879 ...  
880 ...  
881 ...  
882 ...  
883 ...  
884 ...  
885 ...  
886 ...  
887 ...  
888 ...  
889 ...  
890 ...  
891 ...  
892 ...  
893 ...  
894 ...  
895 ...  
896 ...  
897 ...  
898 ...  
899 ...  
900 ...  
901 ...  
902 ...  
903 ...  
904 ...  
905 ...  
906 ...  
907 ...  
908 ...  
909 ...  
910 ...  
911 ...  
912 ...  
913 ...  
914 ...  
915 ...  
916 ...  
917 ...  
918 ...  
919 ...  
920 ...  
921 ...  
922 ...  
923 ...  
924 ...  
925 ...  
926 ...  
927 ...  
928 ...  
929 ...  
930 ...  
931 ...  
932 ...  
933 ...  
934 ...  
935 ...  
936 ...  
937 ...  
938 ...  
939 ...  
940 ...  
941 ...  
942 ...  
943 ...  
944 ...  
945 ...  
946 ...  
947 ...  
948 ...  
949 ...  
950 ...  
951 ...  
952 ...  
953 ...  
954 ...  
955 ...  
956 ...  
957 ...  
958 ...  
959 ...  
960 ...  
961 ...  
962 ...  
963 ...  
964 ...  
965 ...  
966 ...  
967 ...  
968 ...  
969 ...  
970 ...  
971 ...  
972 ...  
973 ...  
974 ...  
975 ...  
976 ...  
977 ...  
978 ...  
979 ...  
980 ...  
981 ...  
982 ...  
983 ...  
984 ...  
985 ...  
986 ...  
987 ...  
988 ...  
989 ...  
990 ...  
991 ...  
992 ...  
993 ...  
994 ...  
995 ...  
996 ...  
997 ...  
998 ...  
999 ...  
9999 ...



# Gitflow

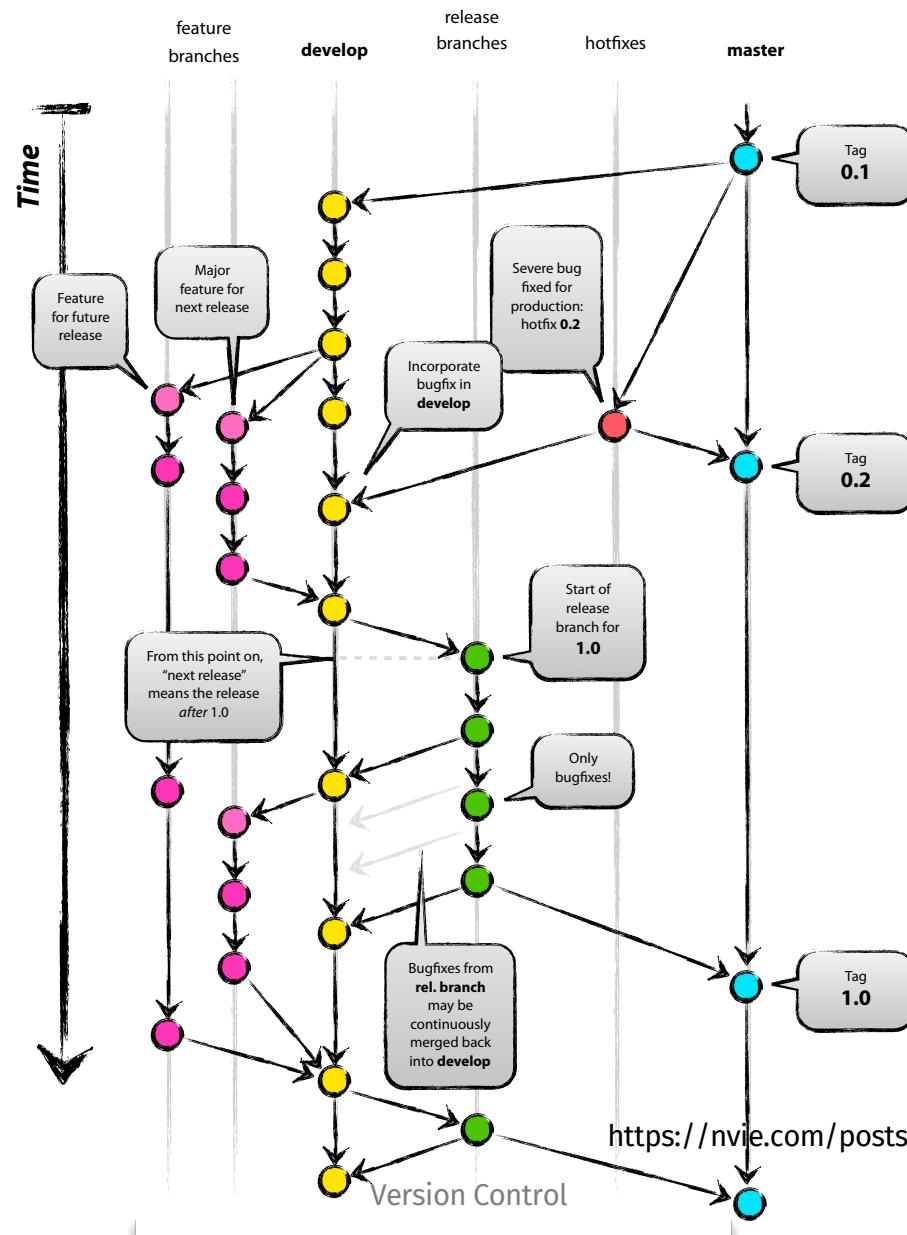


# Gitflow Collaboration

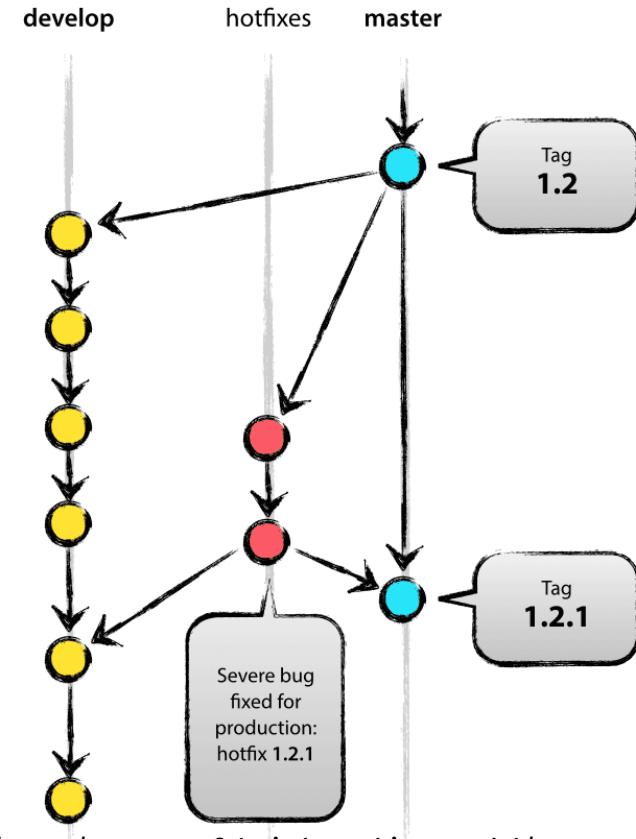
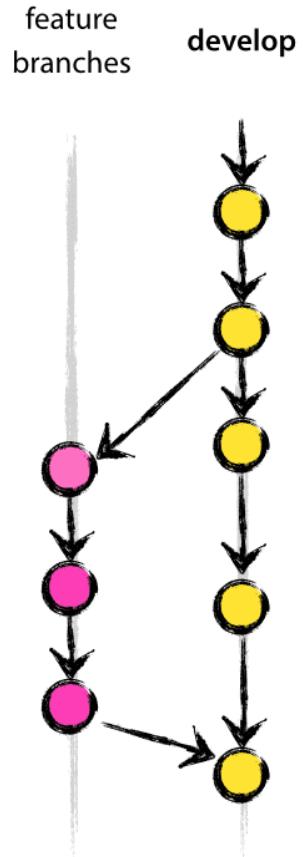
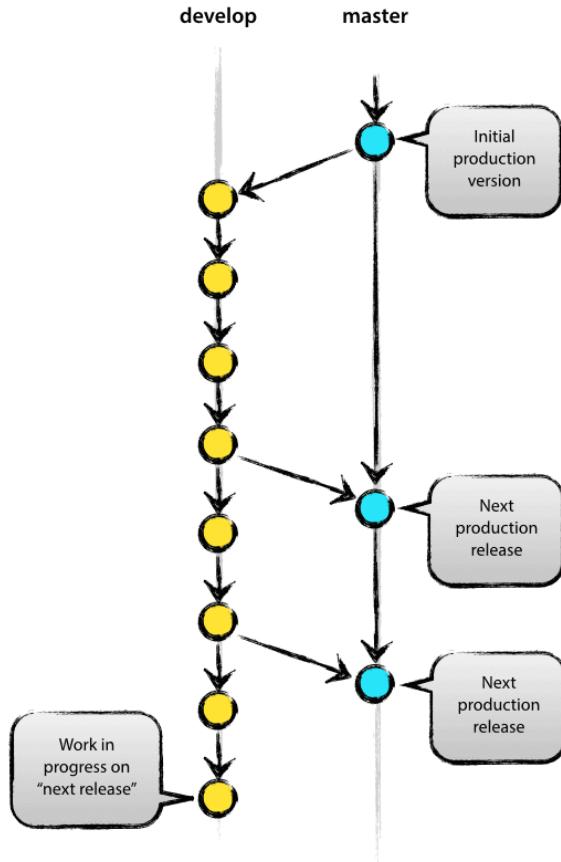


<https://nvie.com/posts/a-successful-git-branching-model/>

# Gitflow



# Gitflow Branching

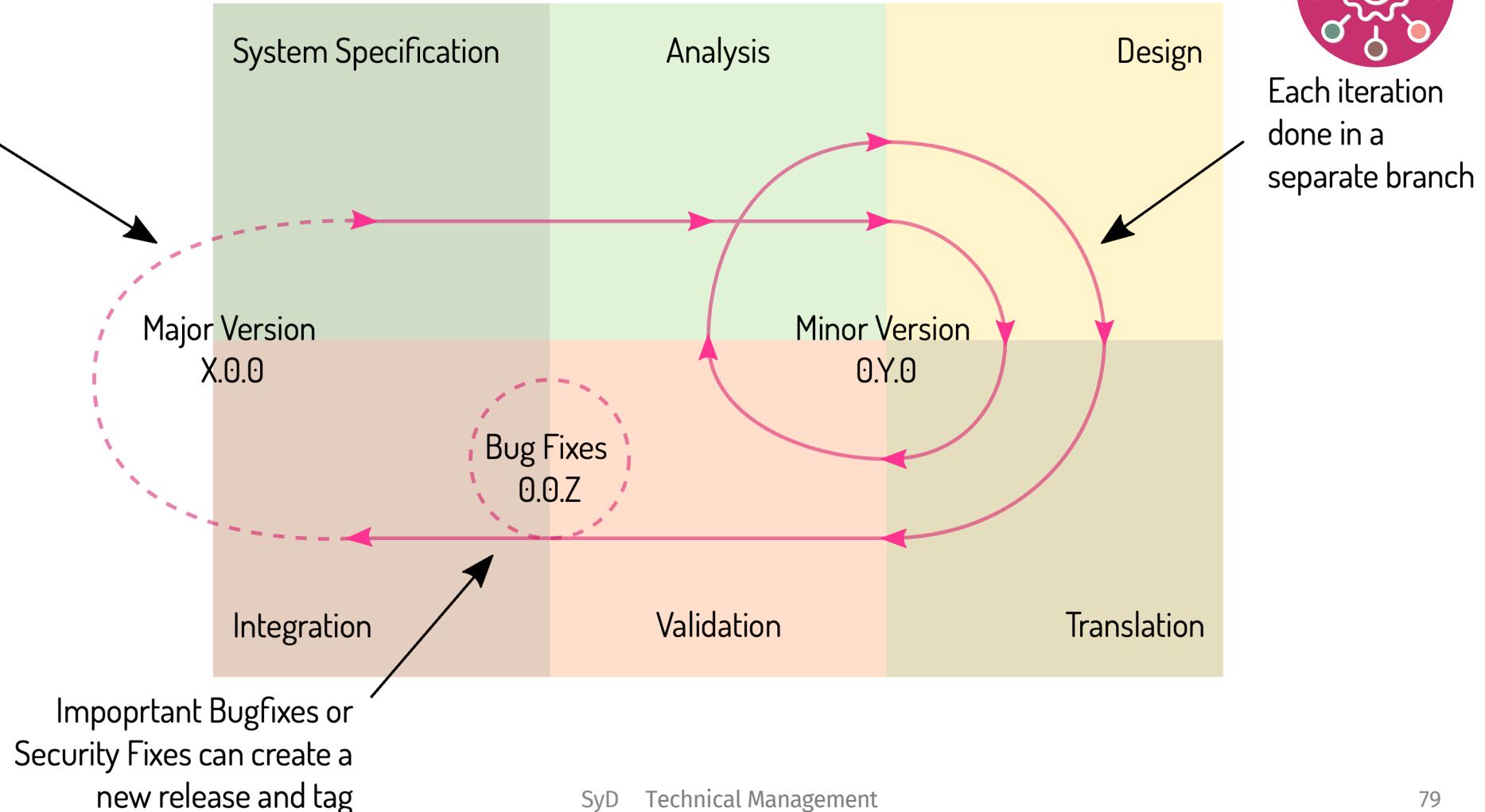


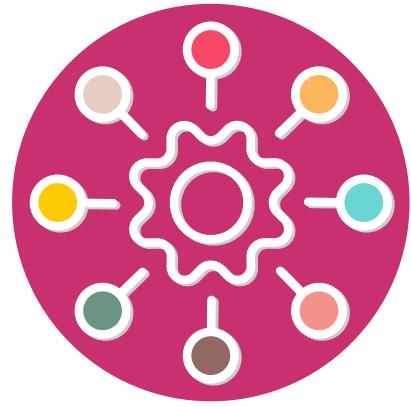
<https://nvie.com/posts/a-successful-git-branching-model/>

# Gitflow vs 6q



Each major version change creates a tag





# Git CI/CD

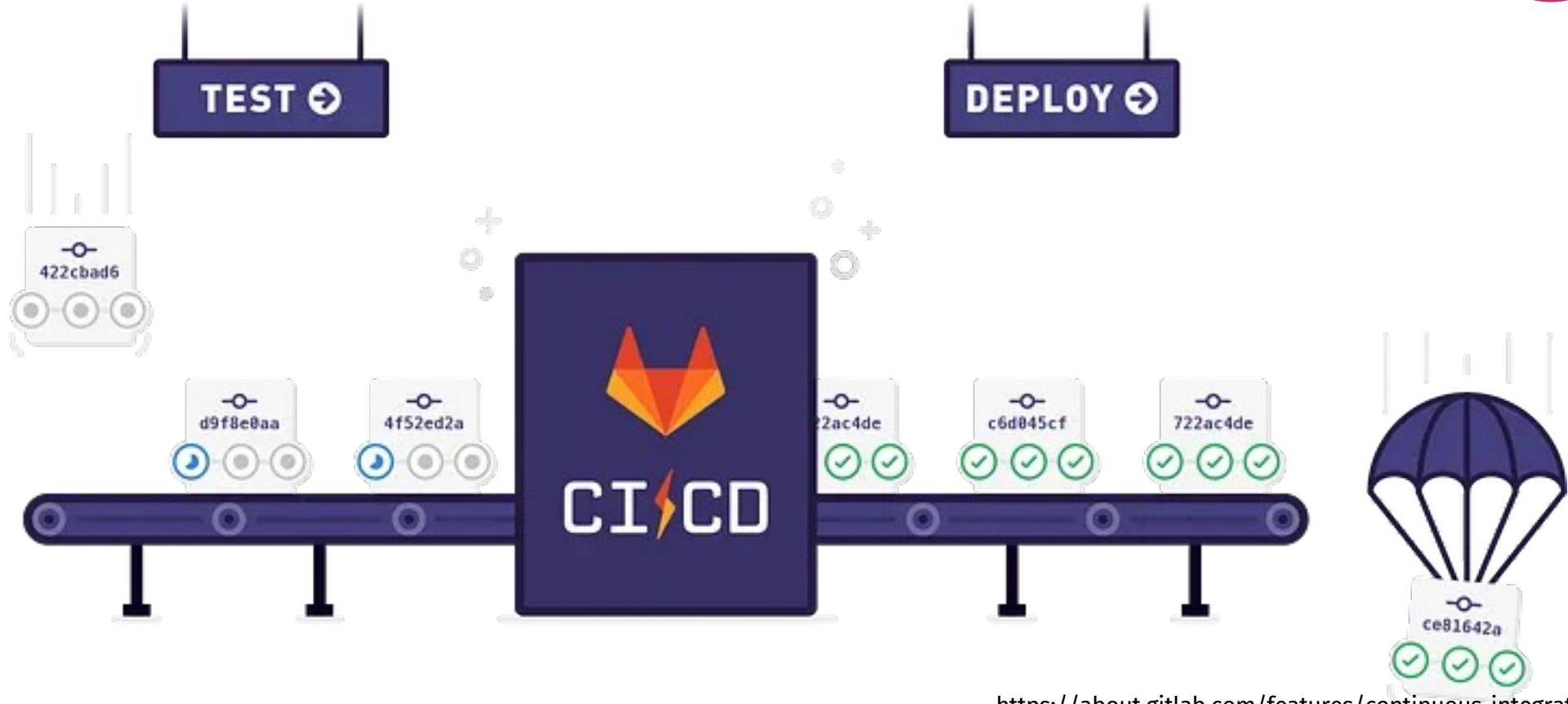
# Qu'est-ce que la CI/CD ?



- L'intégration continue (Continuous Integration, CI) est la pratique qui consiste à intégrer fréquemment des modifications de code dans un référentiel partagé, qui est ensuite automatiquement construit et testé.
- La livraison continue (Continuous Delivery, CD) va encore plus loin en déployant automatiquement les modifications du code dans des environnements de type production afin de les tester et de les valider.
- L'automatisation des tests est un élément essentiel de la CI/CD, car elle permet de détecter les bogues et autres problèmes à un stade précoce du processus de développement.
- Les outils les plus courants sont GitLab CI/CD, Github Actions, Jenkins, CircleCI et Travis CI.

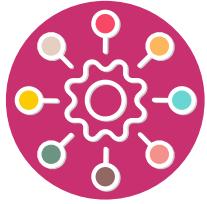


# Qu'est-ce que la CI/CD ?

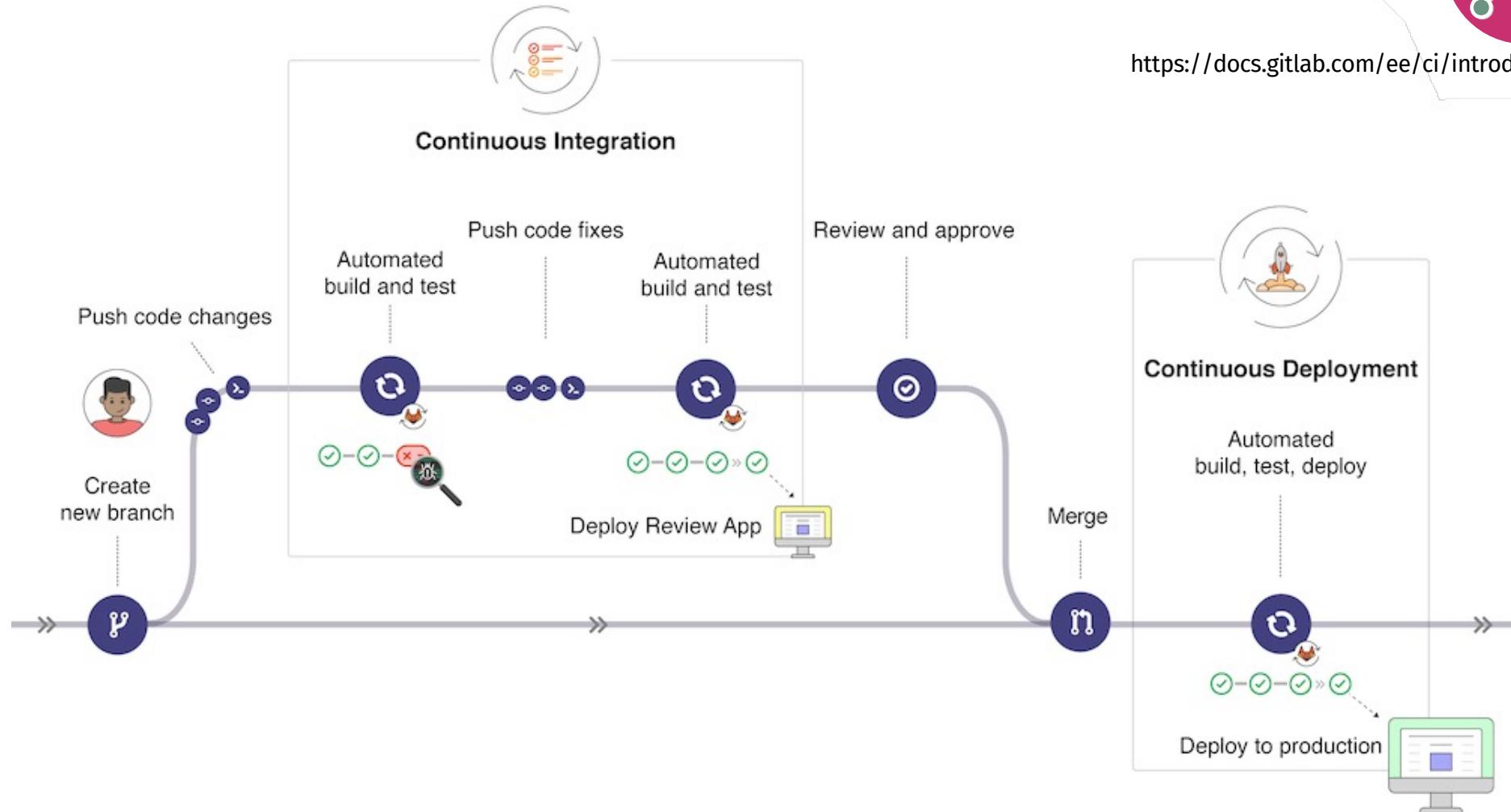


<https://about.gitlab.com/features/continuous-integration/>

# Gitlab Workflow



<https://docs.gitlab.com/ee/ci/introduction/>



# Estimation des tâches

## Estimation 4

Créer un Star Wars  
Star Destroyer (4784pcs)

X Point(s)



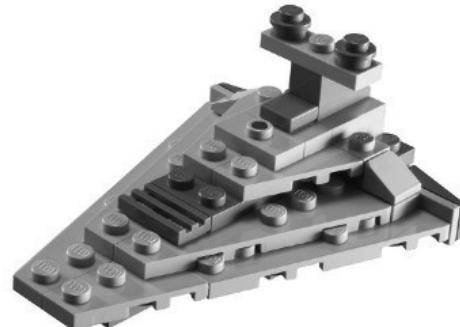
0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	$\infty$
?	☕		

# Estimation des tâches

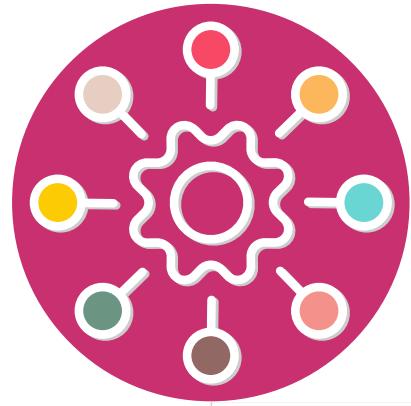
## Estimation 5

Créer un Star Wars  
Star Destroyer (37pcs)

X Point(s)



0	$\frac{1}{2}$	1	2
3	5	8	13
20	40	100	$\infty$
?	☕		



SYSTEMS DESIGN / INTRODUCTION TO GIT - PART A

## Introduction to git - Part A

Installation & Setup



SYSTEMS DESIGN / INTRODUCTION TO GIT - PART B

## Introduction to git - Part B



## Contents

1 Goal .....	1
2 Installation .....	2
3 Markdown .....	5
4 Outro .....	7
A GIT commands .....	8
B Most used Git commands .....	9

## Contents

1 Goals .....	2
2 Outils .....	2
3 Basis Operationen .....	3
4 Branch and Merge .....	11
5 Gitgraph .....	15
6 Gitflow .....	16
7 Extras .....	18