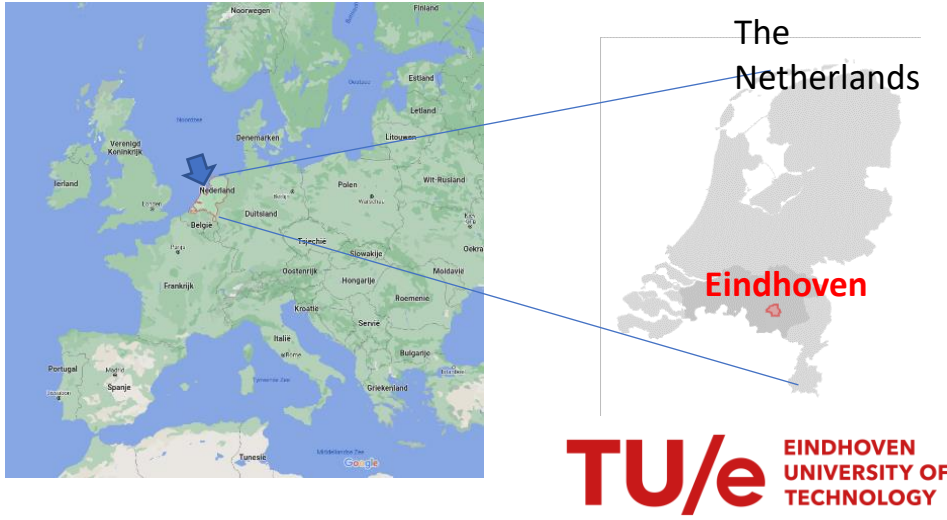


Alessandro Corbetta

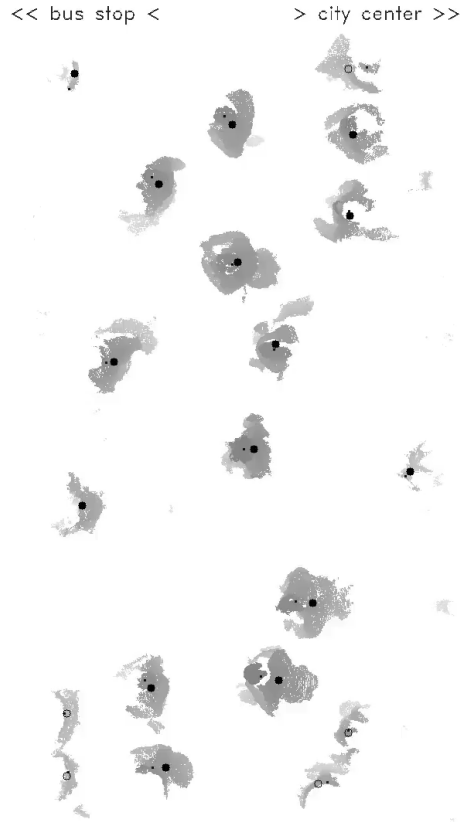
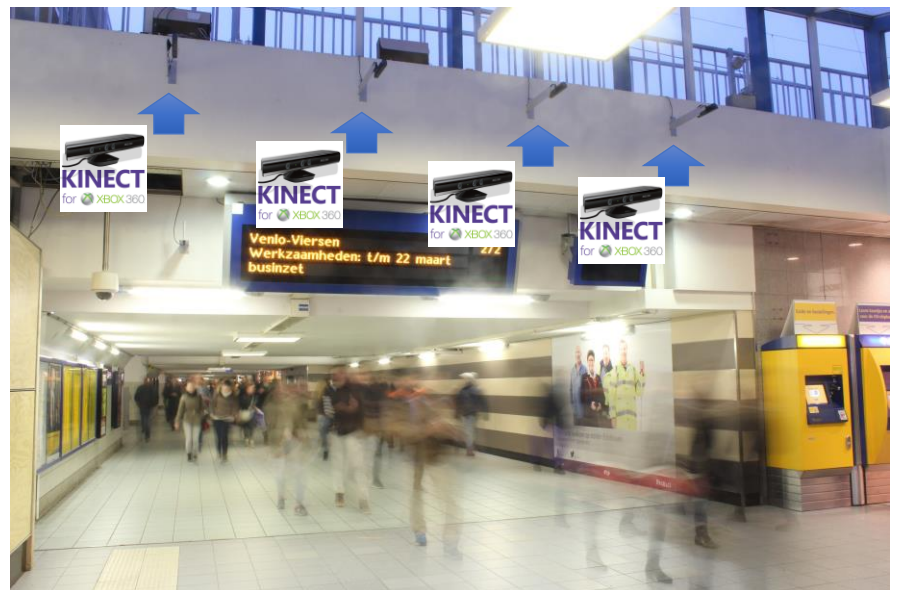
Pedestrian dynamics since 2012
fluid mechanics & stat. phys. Perspective

Machine Learning for fluid mechanics since 2016



Department of Applied Physics
Fluids and Flows

<https://crowdflow.phys.tue.nl>



Alessandro Corbetta



📁 **1,258**
PROJECTS

New project

[View latest projects >](#)

👤 **317** ⓘ
USERS [Users Statistics](#)

New user

[View latest users >](#)

👥 **46**
GROUPS

New group

[View latest groups >](#)



Computing cluster



Virtualization cluster, Hyperconverged storage

Coding: collaborative,
trustworthy, reproducible

Alessandro Corbetta

An infrastructure in shells

Development &
Research process
management

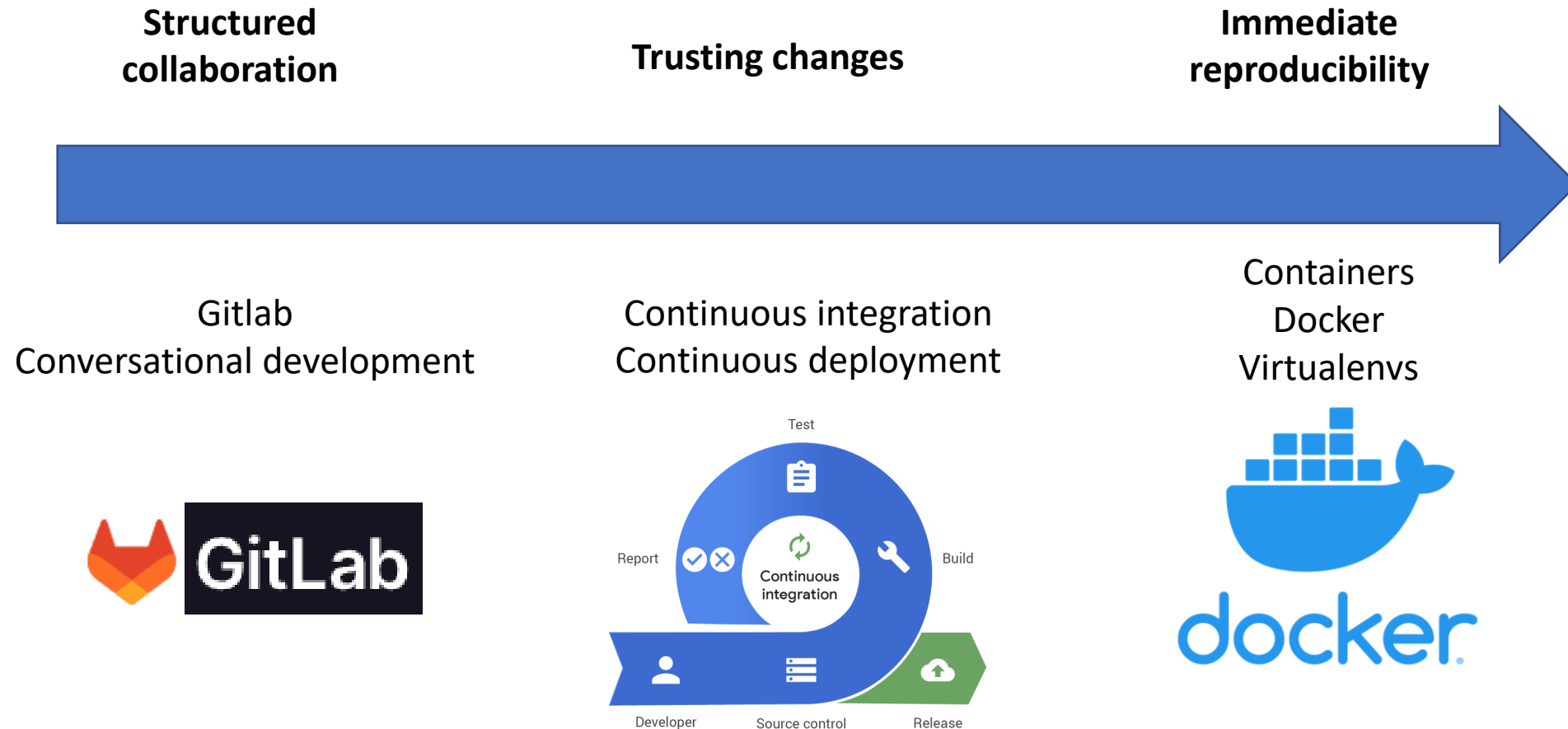
Process & high-density data
vault



Accessibility,
Open source &
Community development

Containerization:
bitwise reproducibility

Plan of the lectures



Git

Startup



1 project



1 git repository



Linus Torvalds, 2005
(frustrated by bitkeeper)



Repo examples:

- 1 code project



- 1 paper



- 1 student



- 1 project proposal



- 1 presentation





```
cd project_folder  
git init .
```

Next slides: 8 basic commands

Tracking versions


Make a snapshot of the current version

 \$ git **add** file1.py file2.py

 \$ git **commit** -m 'added files'

And so on for the next snapshot

 \$ git add file1.py

 \$ git commit -m 'changed XX YY'

Changes

git **diff**

```
macorbe@Alessandros-MacBook-Pro my_code_2 [master] $ git diff
diff --git a/converter.py b/converter.py
index 4511134..58edb46 100644
--- a/converter.py
+++ b/converter.py
@@ -1,13 +1,16 @@
+from __future__ import print_function
 import numpy as np
+import sys
+
 def binary_number_string_parser(inp
     , datatype = float):
-
+ print (inp)
 return 0

 def main():
-
- return binary_number_string_parser(0)
+
+ return binary_number_string_parser(sys.argv[1])

 if __name__ == '__main__':
 macorbe@Alessandros-MacBook-Pro my_code_2 [master] $
```

Removed content

Added content

Changes

git log

```
my_code_2 —
[acorbe@Alessandros-MacBook-Pro my_code_2 [master] $ git add conv
[acorbe@Alessandros-MacBook-Pro my_code_2 [master] $ git commit -r
[master 96bf846] added command line capturing
 1 file changed, 6 insertions(+), 3 deletions(-)
[acorbe@Alessandros-MacBook-Pro my_code_2 [master] $ git log
commit 96bf846d46db1177cd04613cbb9dbdd71dd1d6f7 (HEAD -> master)
Author: Alessandro Corbetta <corbisoft@gmail.com>
Date: Tue May 1 16:39:02 2018 +0430

    added command line capturing

commit ac1408245b78a7a8b53f5040b8780c1733fdf6af
Author: Alessandro Corbetta <corbisoft@gmail.com>
Date: Tue May 1 16:27:09 2018 +0430

    initial commit
acorbe@Alessandros-MacBook-Pro my_code_2 [master] $ █
```

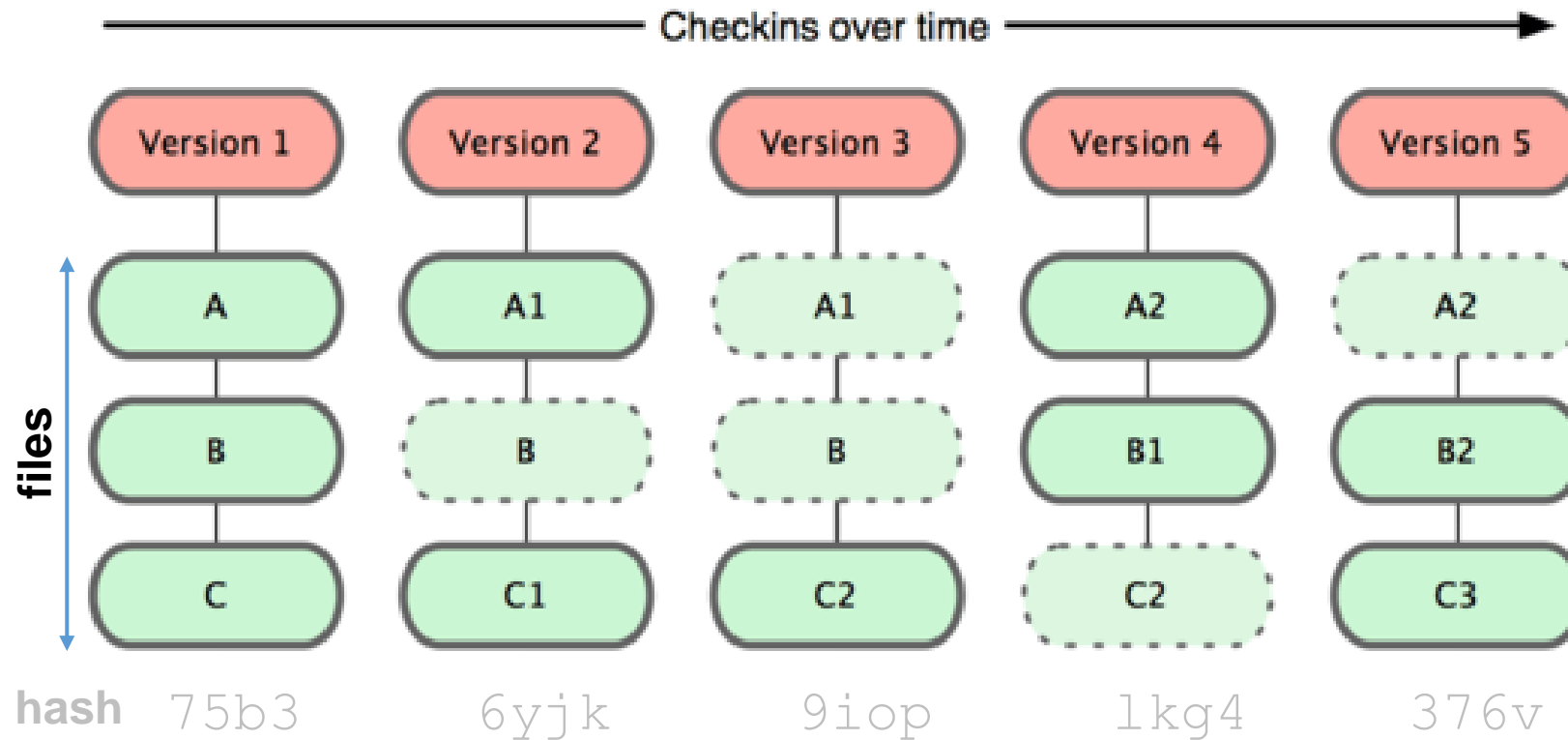
Second snapshot (commit)

First snapshot (commit)

History & rollback

- Each commit holds a complete snapshot of the repository

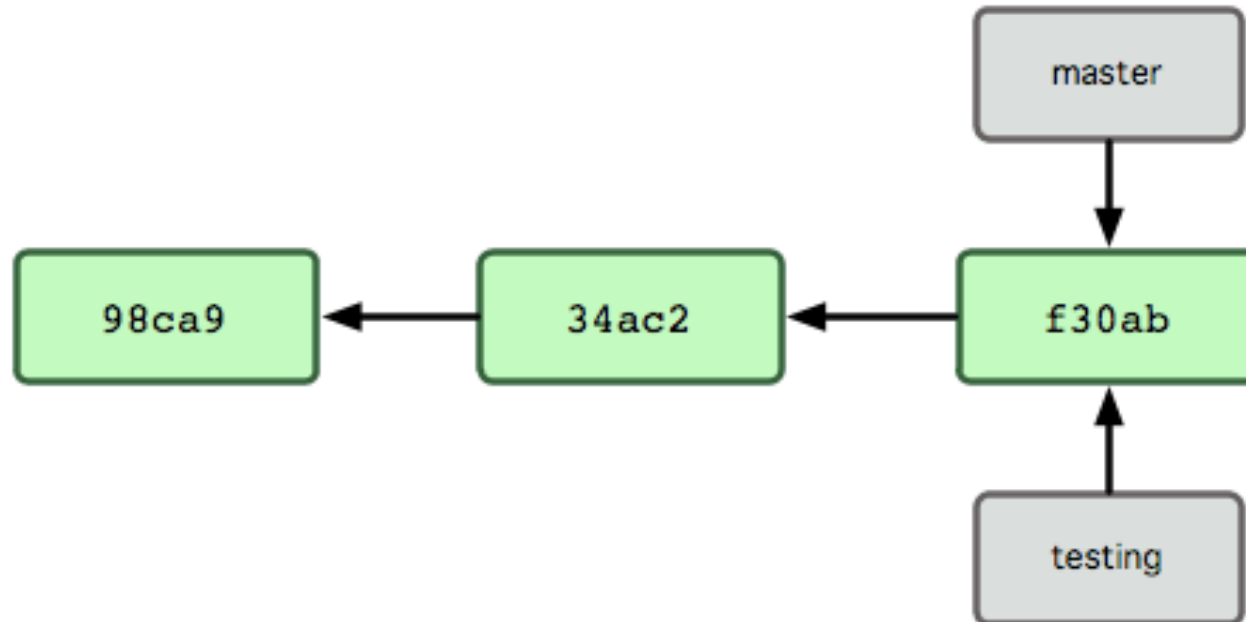
```
git checkout <hash>
```



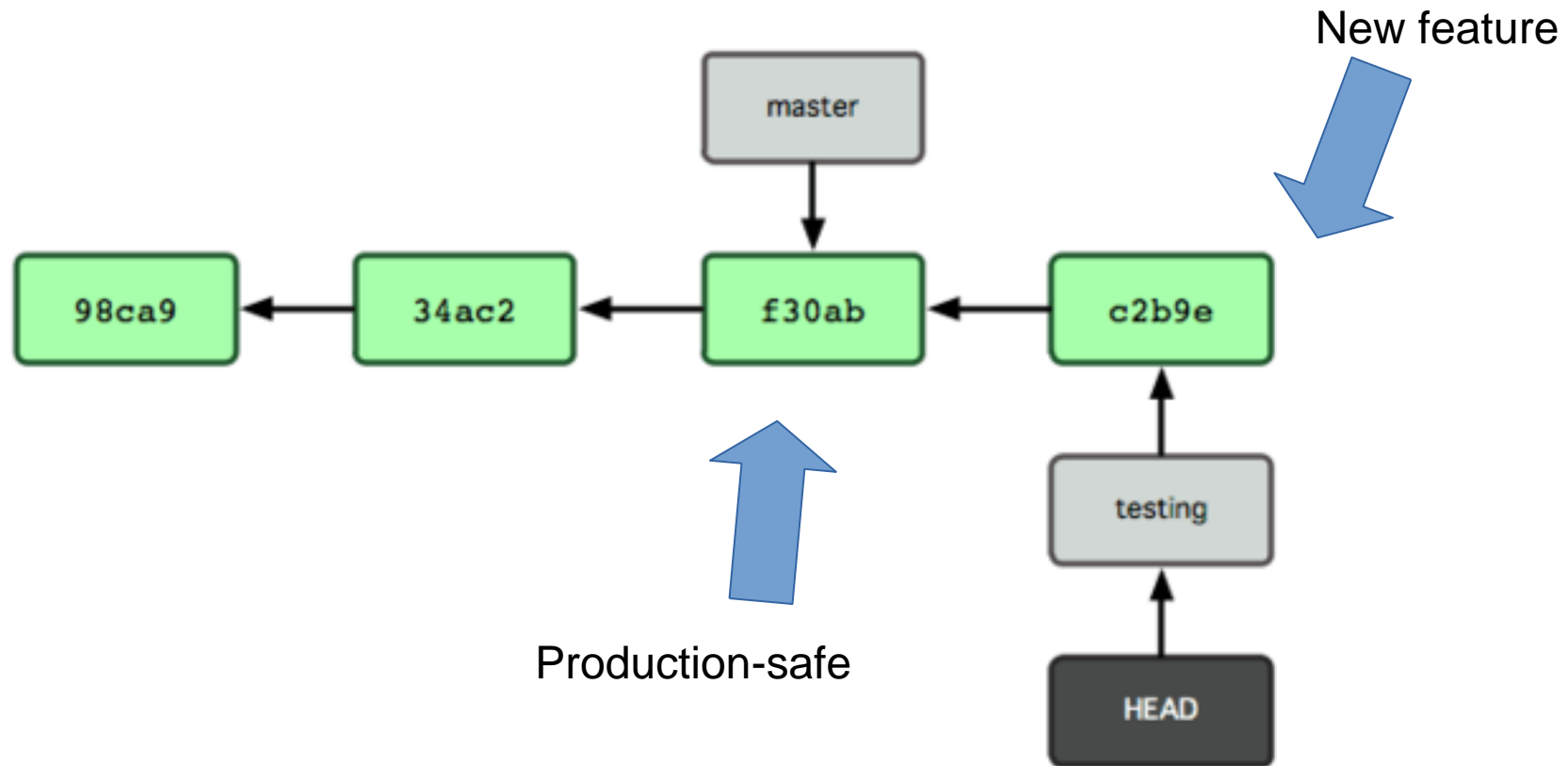
Branches for a safe future

Default branch “master” or “main”

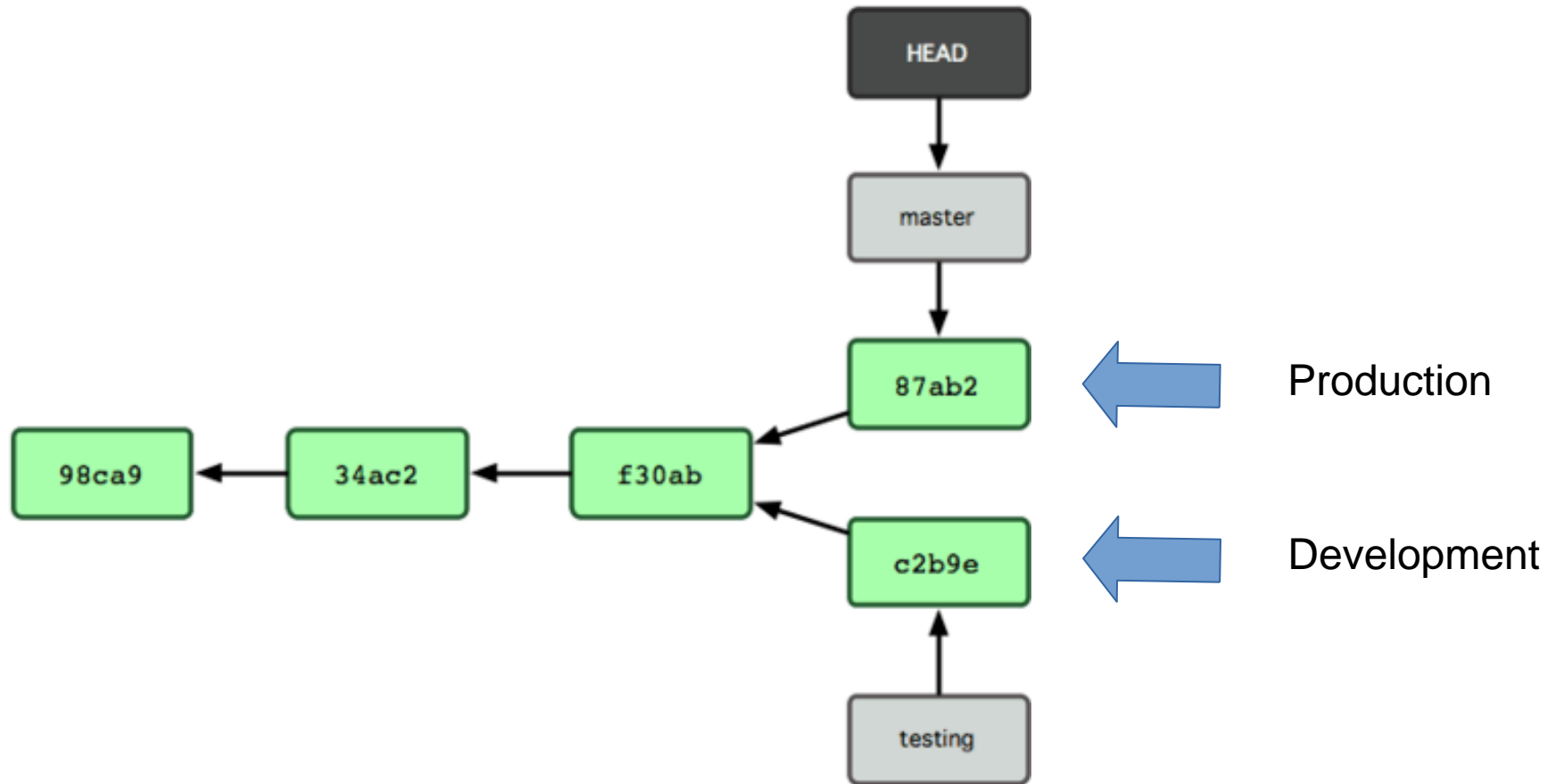
New branch: `git branch testing`



Development in “testing”

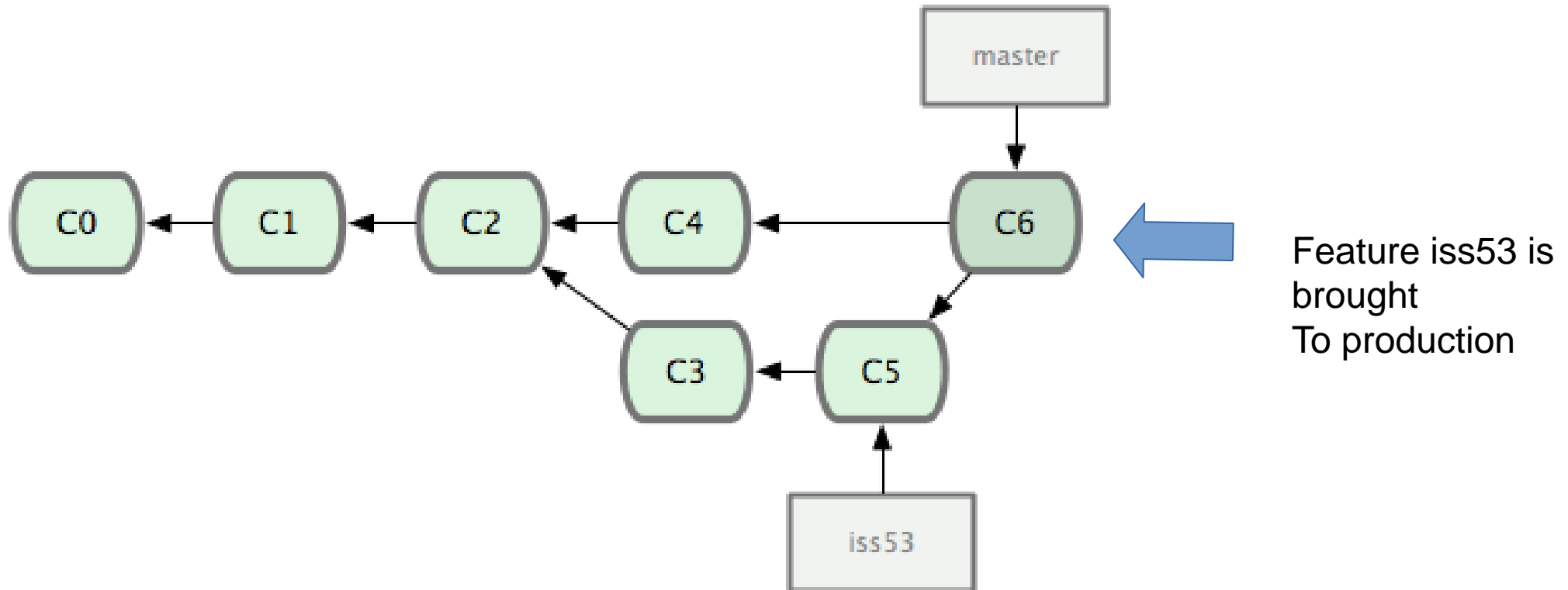


Branches can evolve independently!



Merging

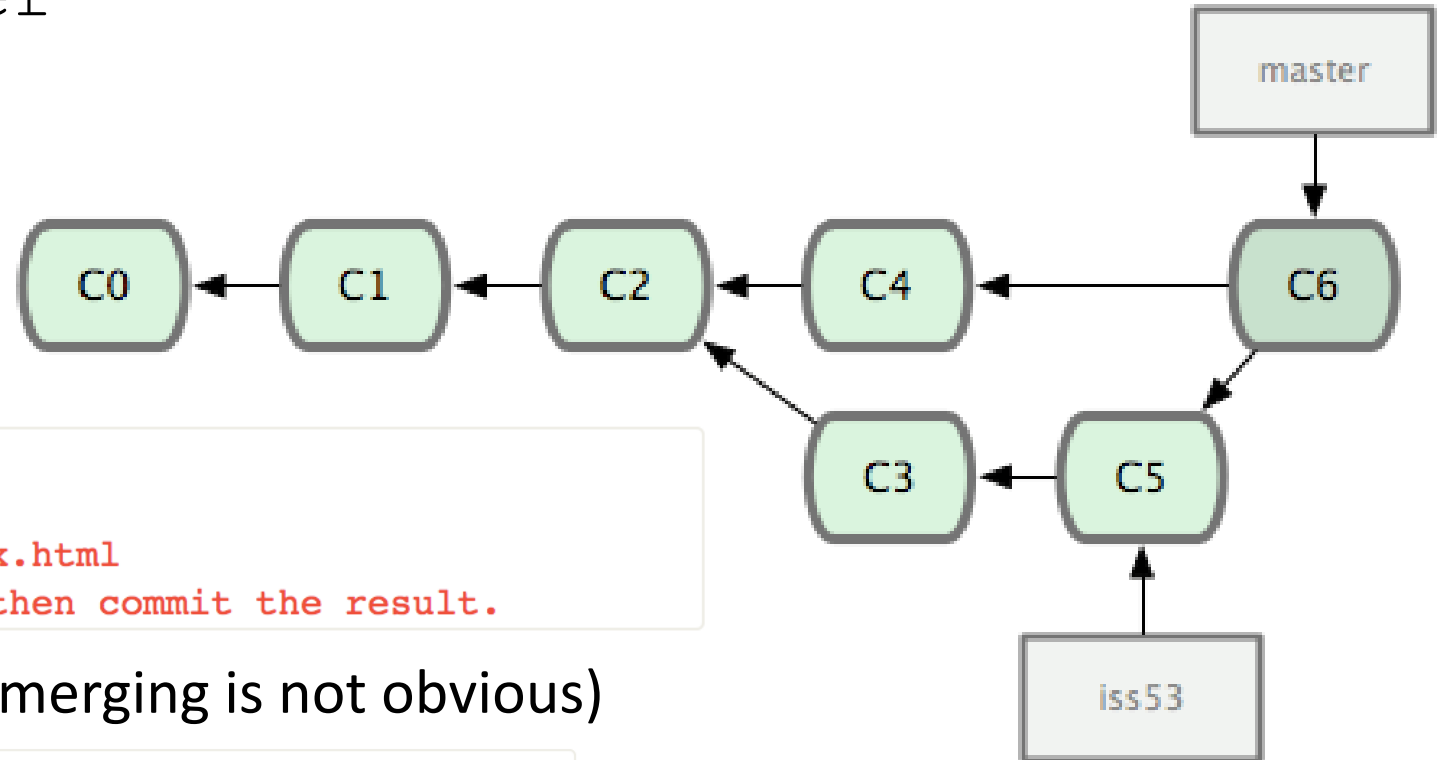
- `git checkout master`
- `git merge iss53`



Merge **conflicts**

certain decisions are for humans only

- `git checkout master`
- `git merge iss53`



```
$ git merge iss53
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

Manual merging might be needed (if merging is not obvious)

```
<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> iss53:index.html
```

Git: basic version control

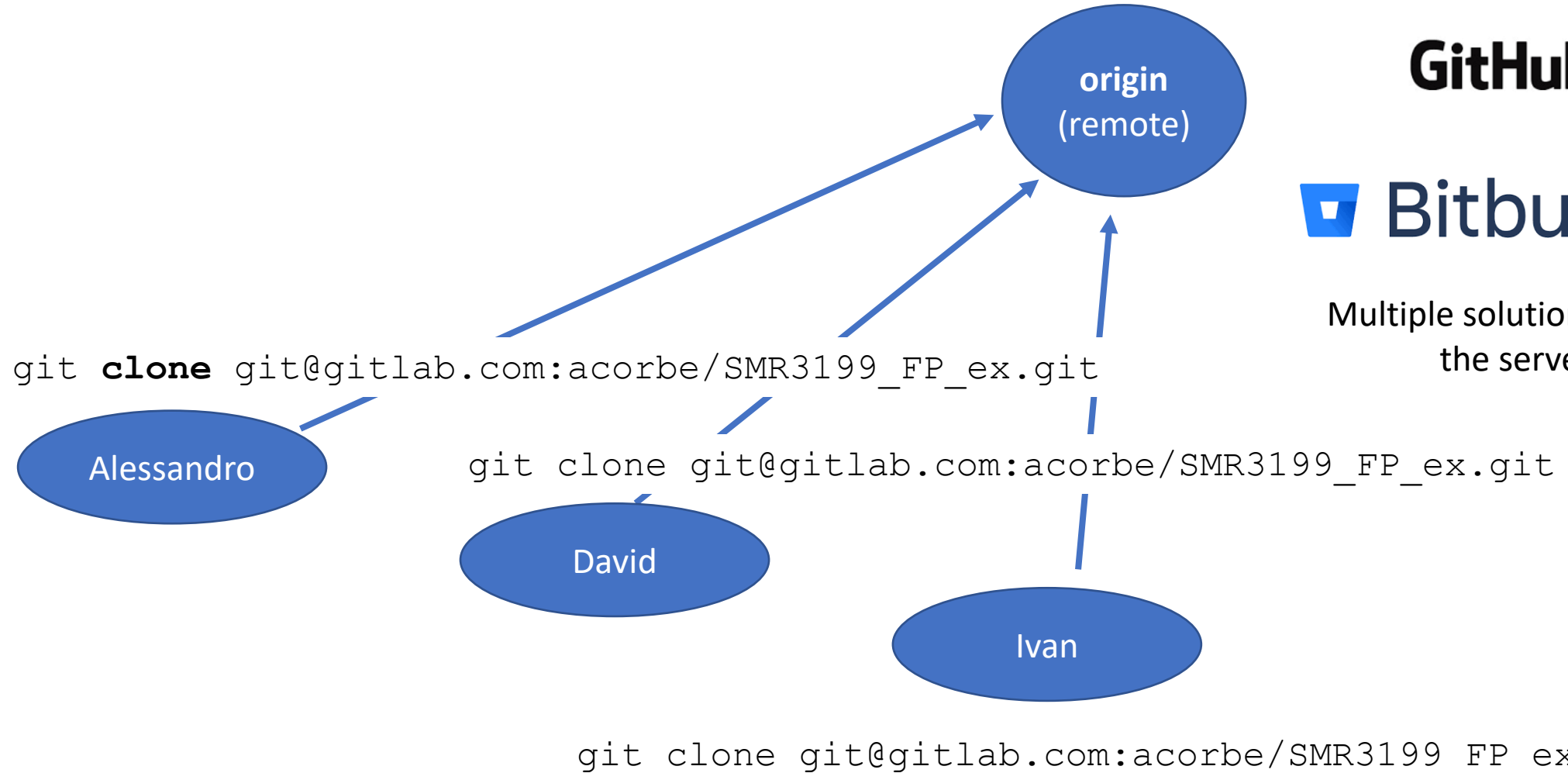
- git init
- git add stage file for snapshot
- git commit take snapshot
- git diff compare snapshots
- git log snapshot list
- git checkout move to a different snapshot
- git branch branch the future
- git merge join different futures in one

Coding: collaborative,
trustworthy, reproducible

Alessandro Corbetta

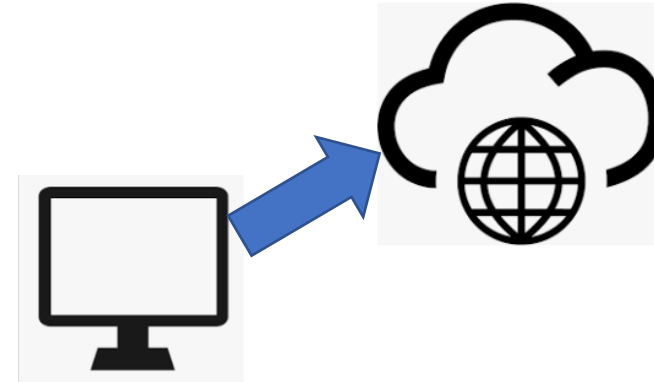
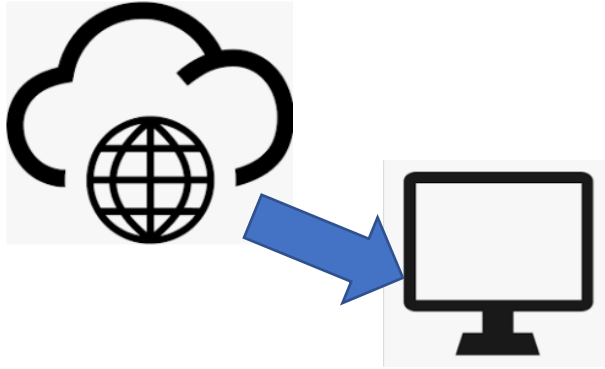
Collaborative development

Users have local repositories mirroring a remote **origin**.



Multiple solutions to host
the server yourself

Pulling/Pushing commits to the origin



To update <branch> locally

- `git checkout <branch>`
- `git pull origin <branch>`

To push to remote

- `git checkout <branch>`
- `git push origin <branch>`

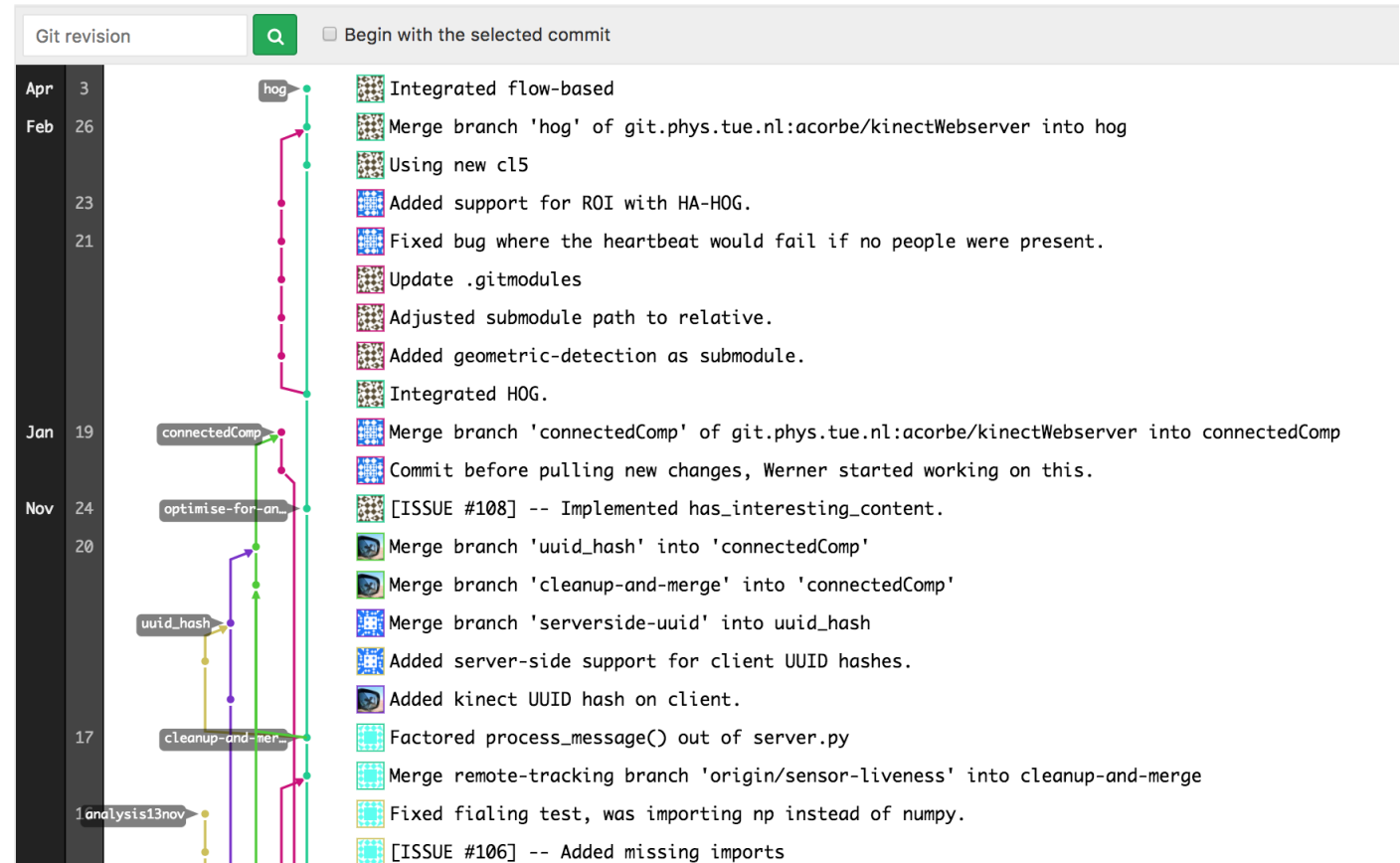
Note: remotes can be **public** or **private**

Pulling/Pushing commits to the origin



Alessandro C > kinectWebserver > Commits

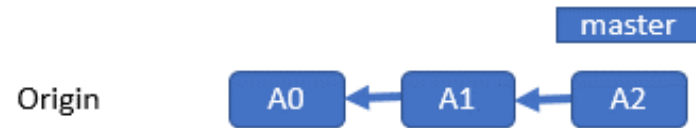
- hog kinectWebserver
- 03 Apr, 2018 1 commit
- Integrated flow-based**
Werner Kroneman authored 4 weeks ago
- 26 Feb, 2018 2 commits
- Merge branch 'hog' of git.phys.tue.nl:acorbe/kinectWebserver into hog**
Werner Kroneman authored 2 months ago
 - Using new cl5**
Werner Kroneman authored 2 months ago
- 23 Feb, 2018 1 commit
- Added support for ROI with HA-HOG.**
Werner Kroneman authored 2 months ago
- 21 Feb, 2018 5 commits
- Fixed bug where the heartbeat would fail if no people were present.**
Werner Kroneman authored 2 months ago
 - Update .gitmodules**
Werner Kroneman authored 2 months ago
 - Adjusted submodule path to relative.**
Werner Kroneman authored 2 months ago



Interaction between two or more developers

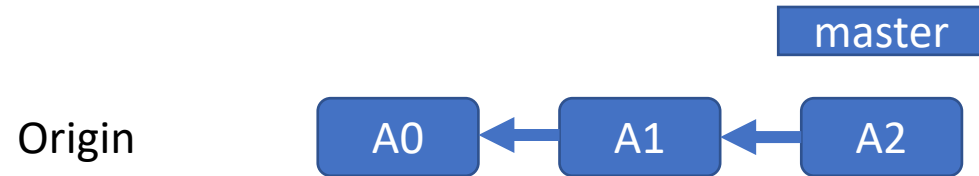
Collaborative development

What happens after clone?



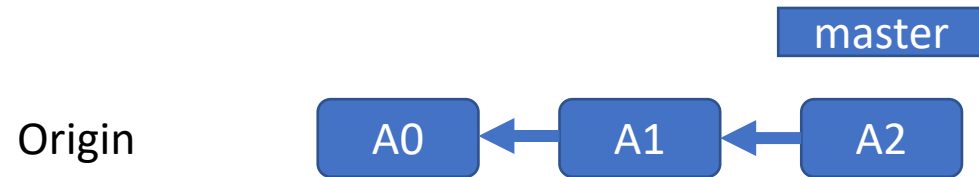
Collaborative development

What happens after clone?



Collaborative development

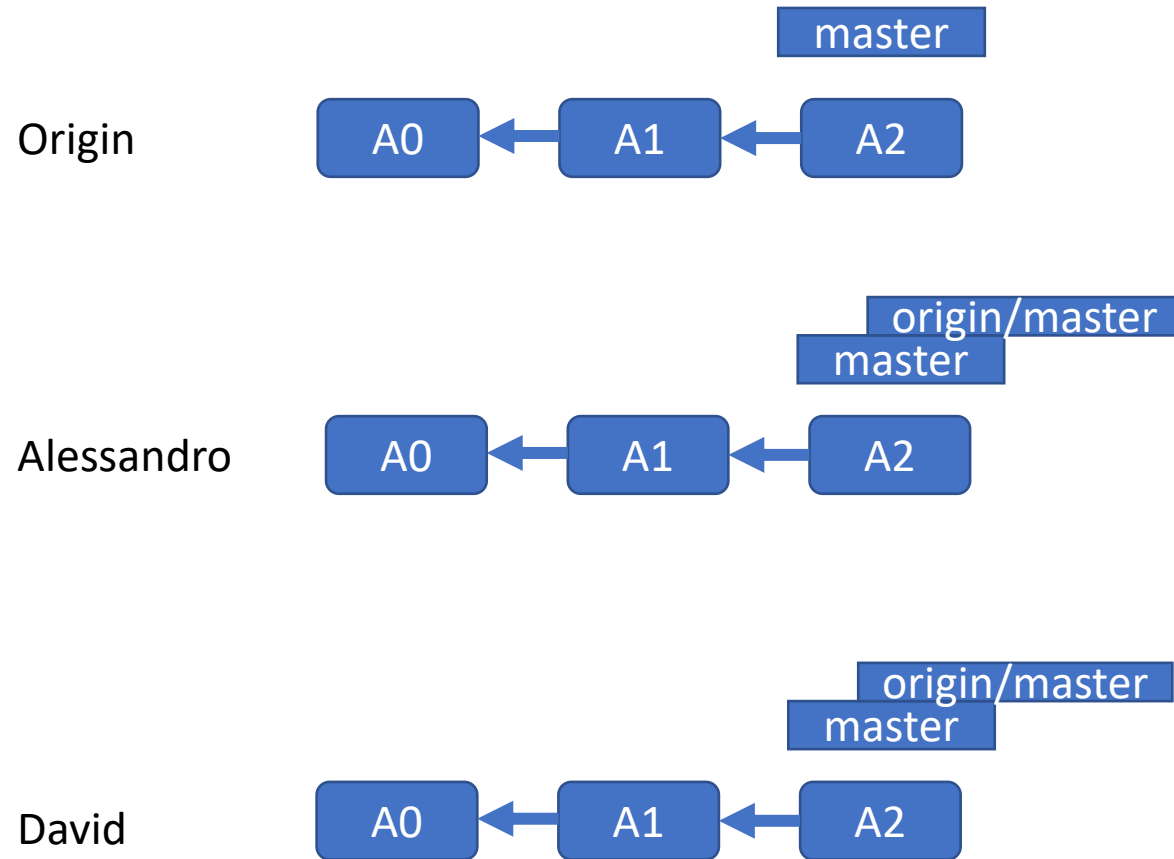
What happens after clone?



Alessandro & David both do `git clone`

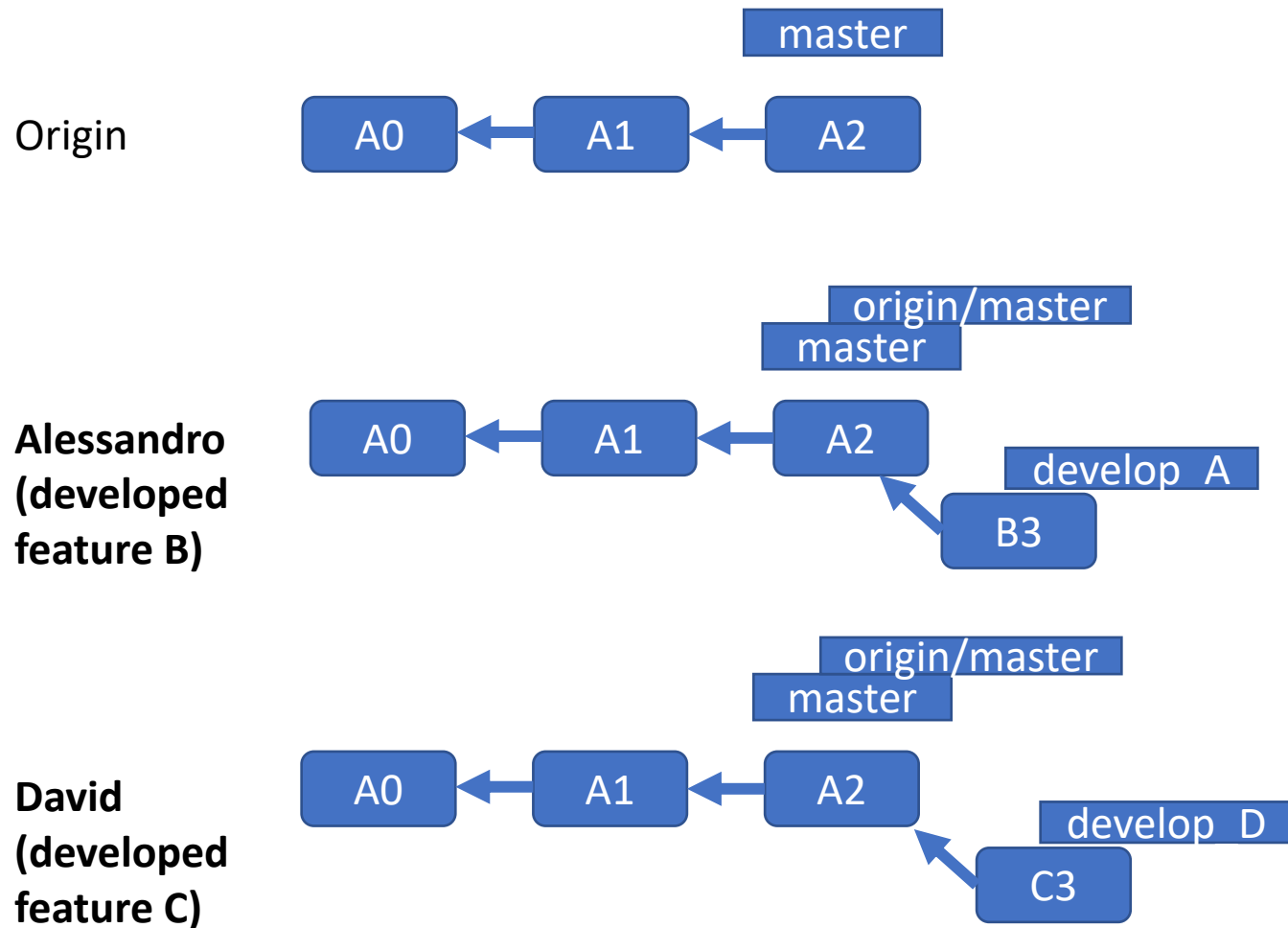
Collaborative development

What happens after clone?



Collaborative development

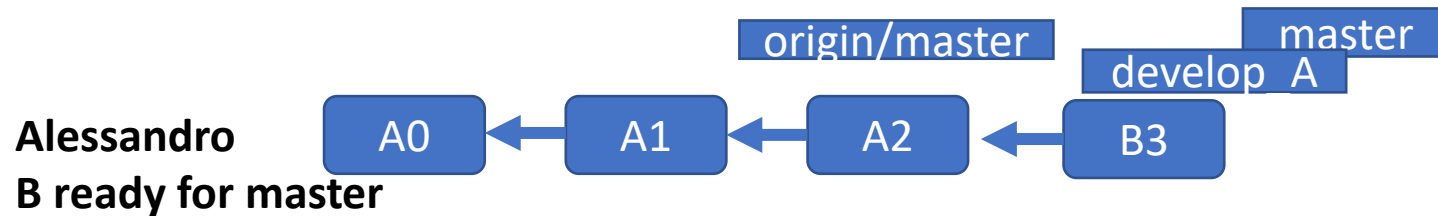
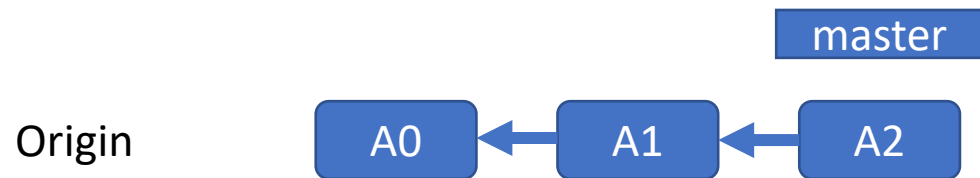
What happens after clone?



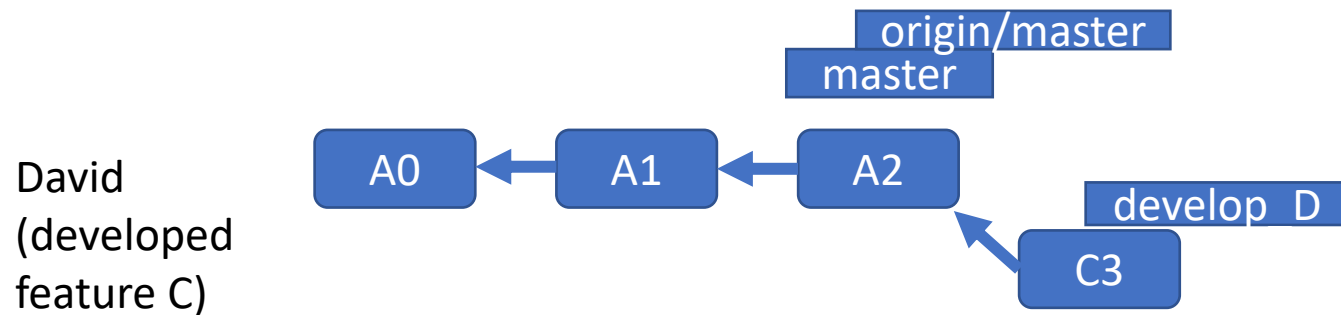
Both David and Alessandro committed

Collaborative development

What happens after clone?

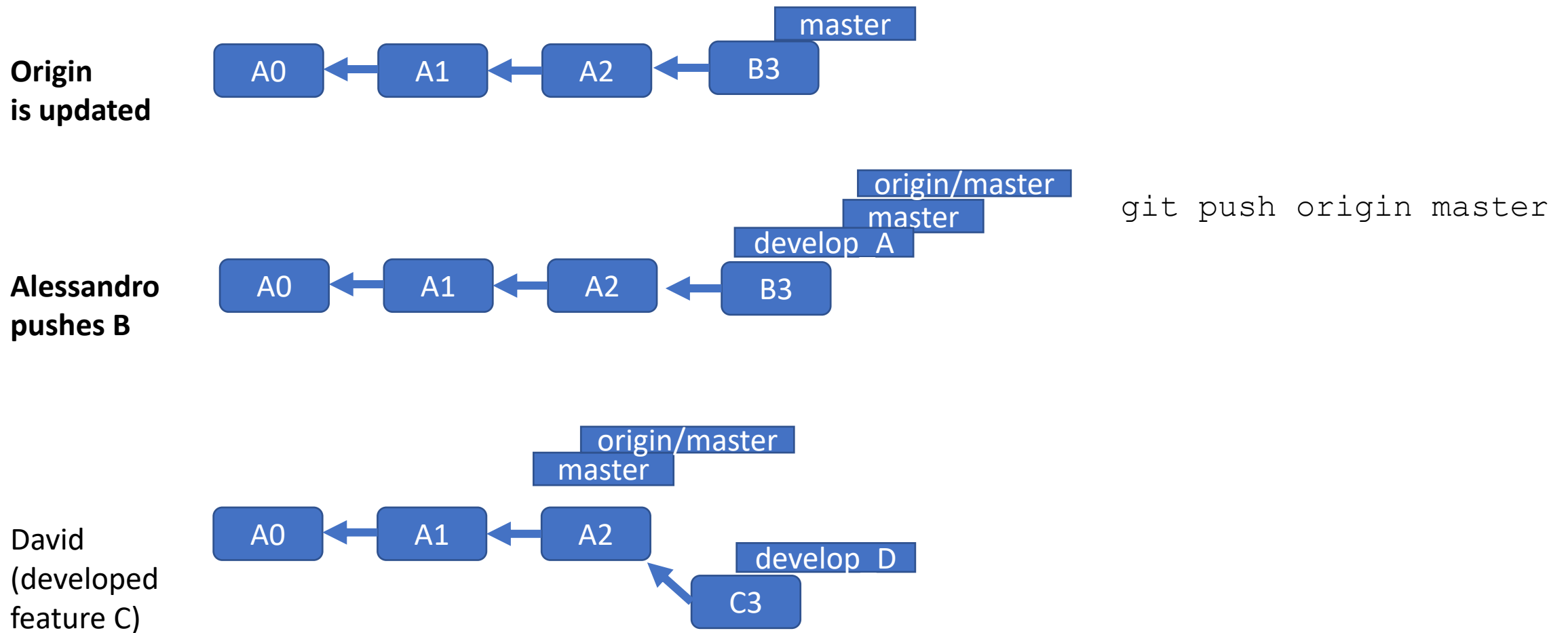


```
git checkout master  
git pull  
git merge develop_A
```



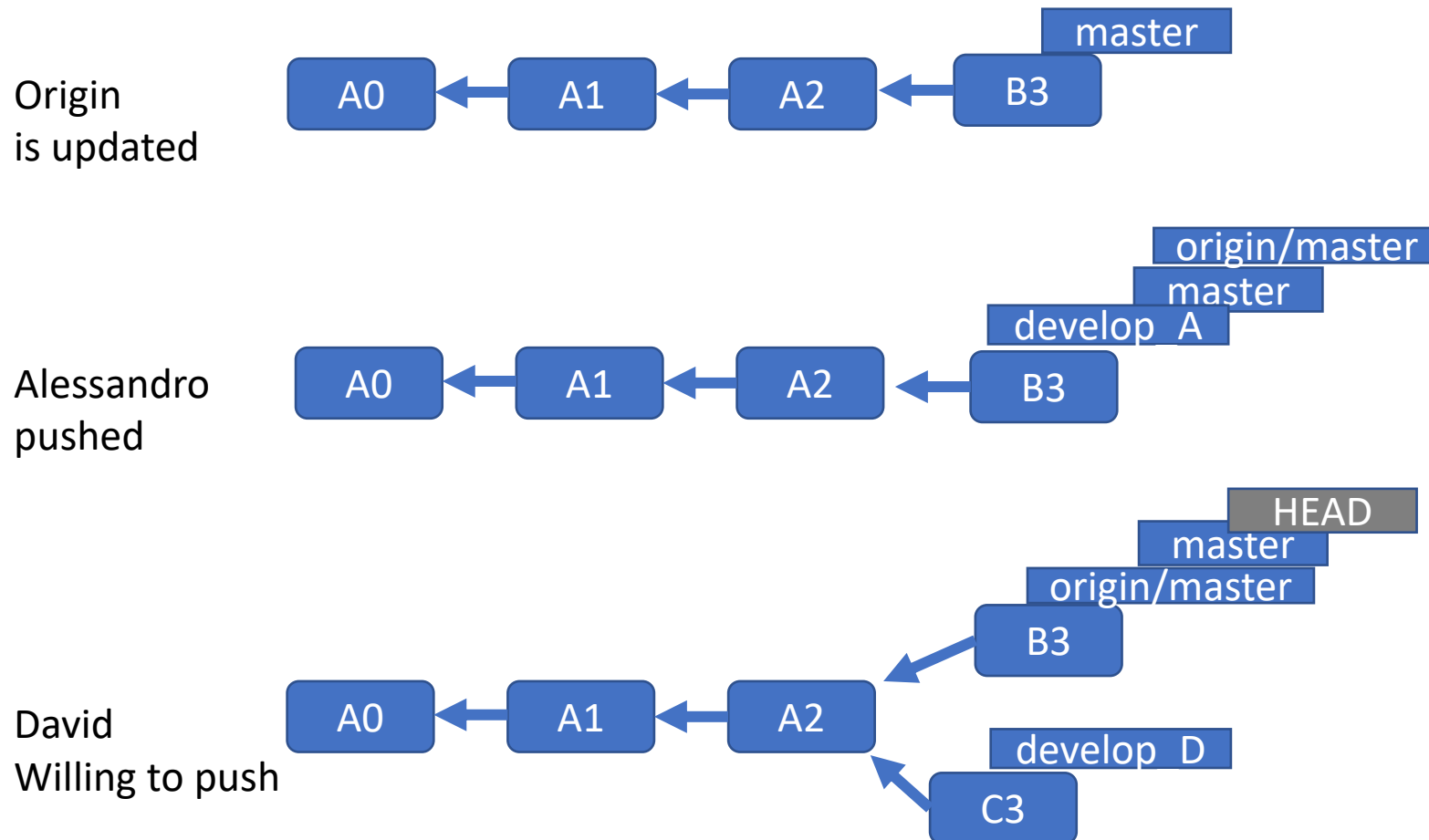
Collaborative development

What happens after clone?



Collaborative development

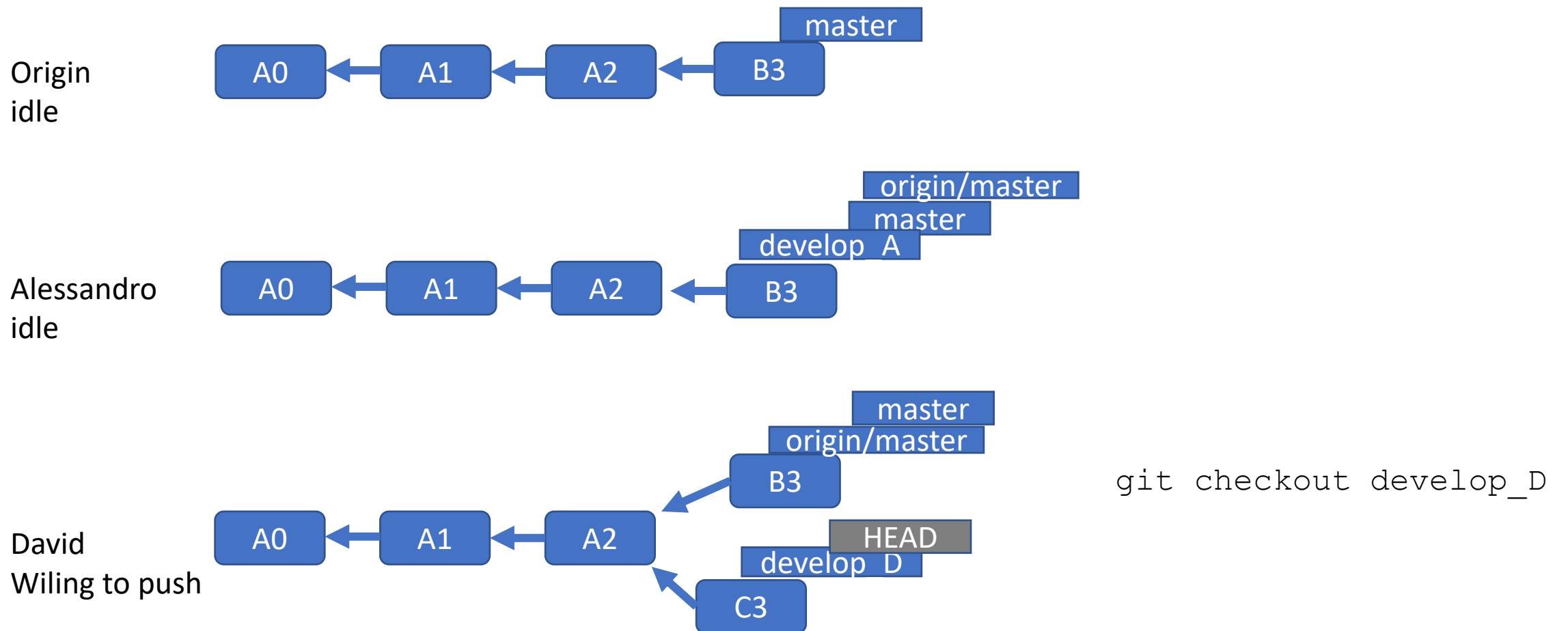
What happens after clone?



```
git checkout master  
git pull #UPDATES!!
```

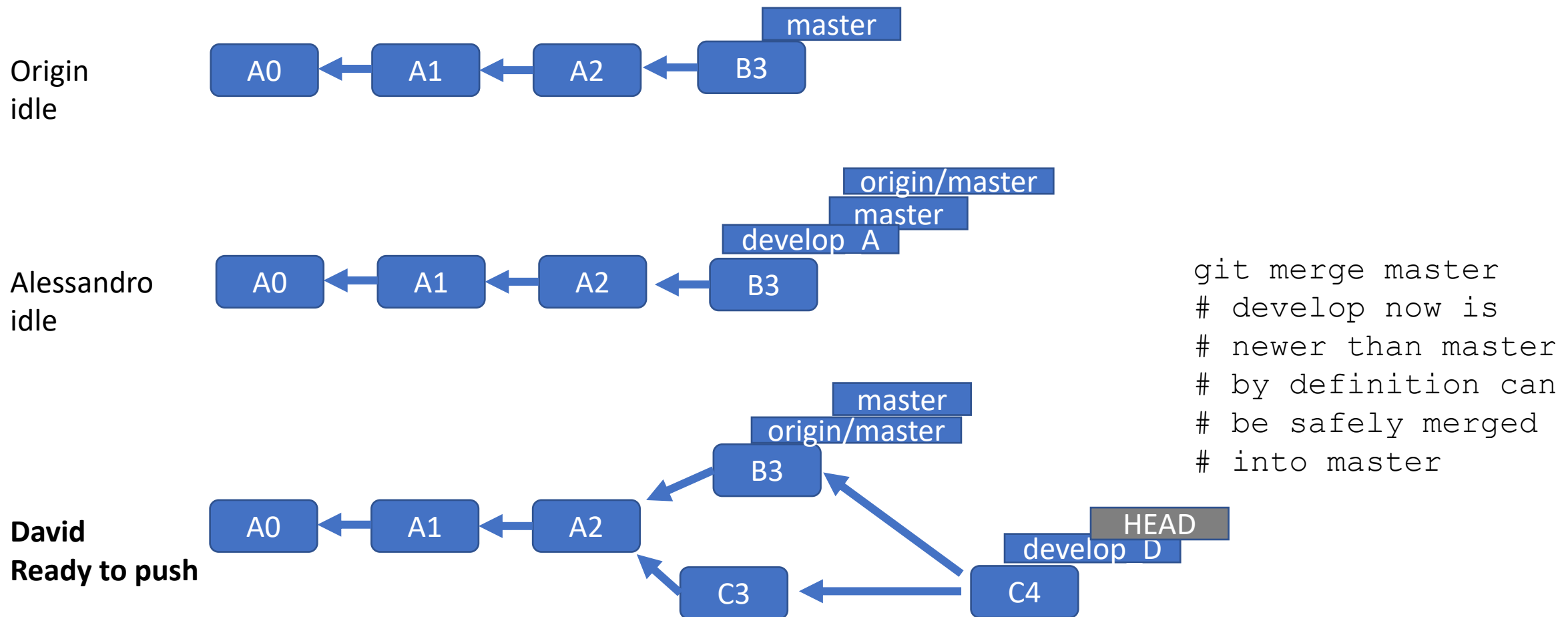
Collaborative development

What happens after clone?



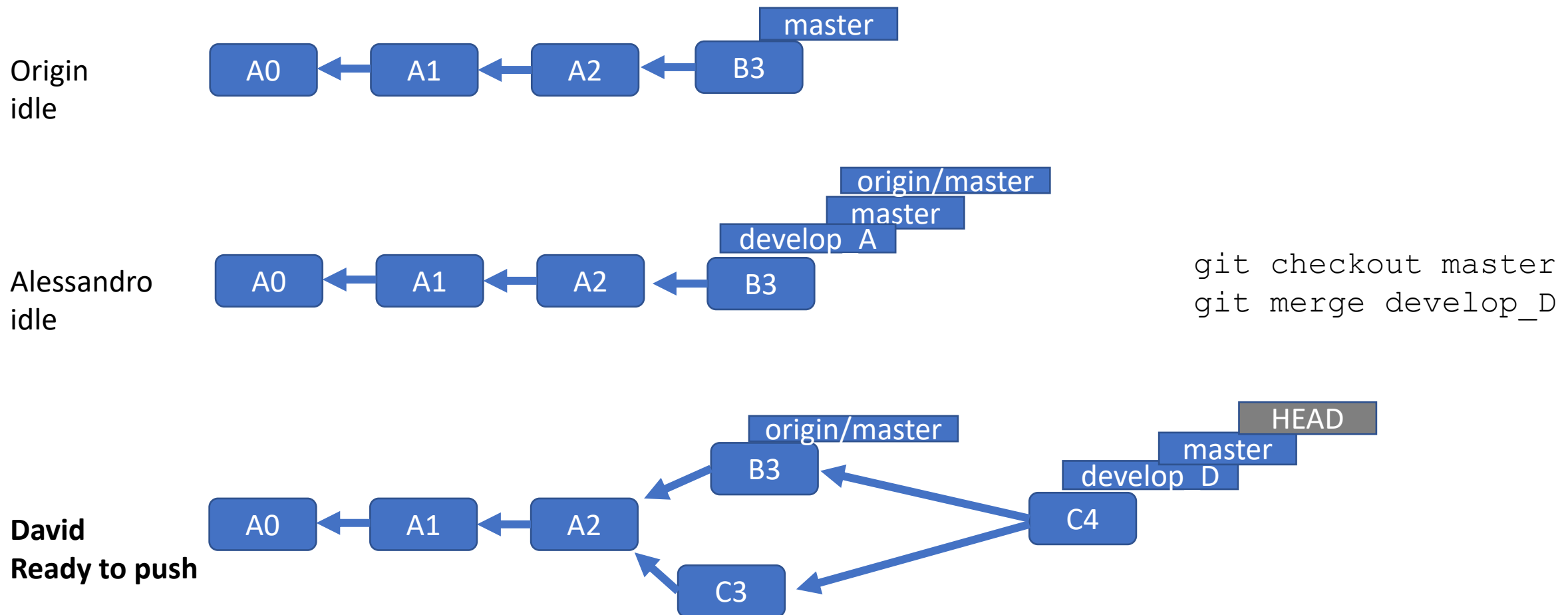
Collaborative development

What happens after clone?



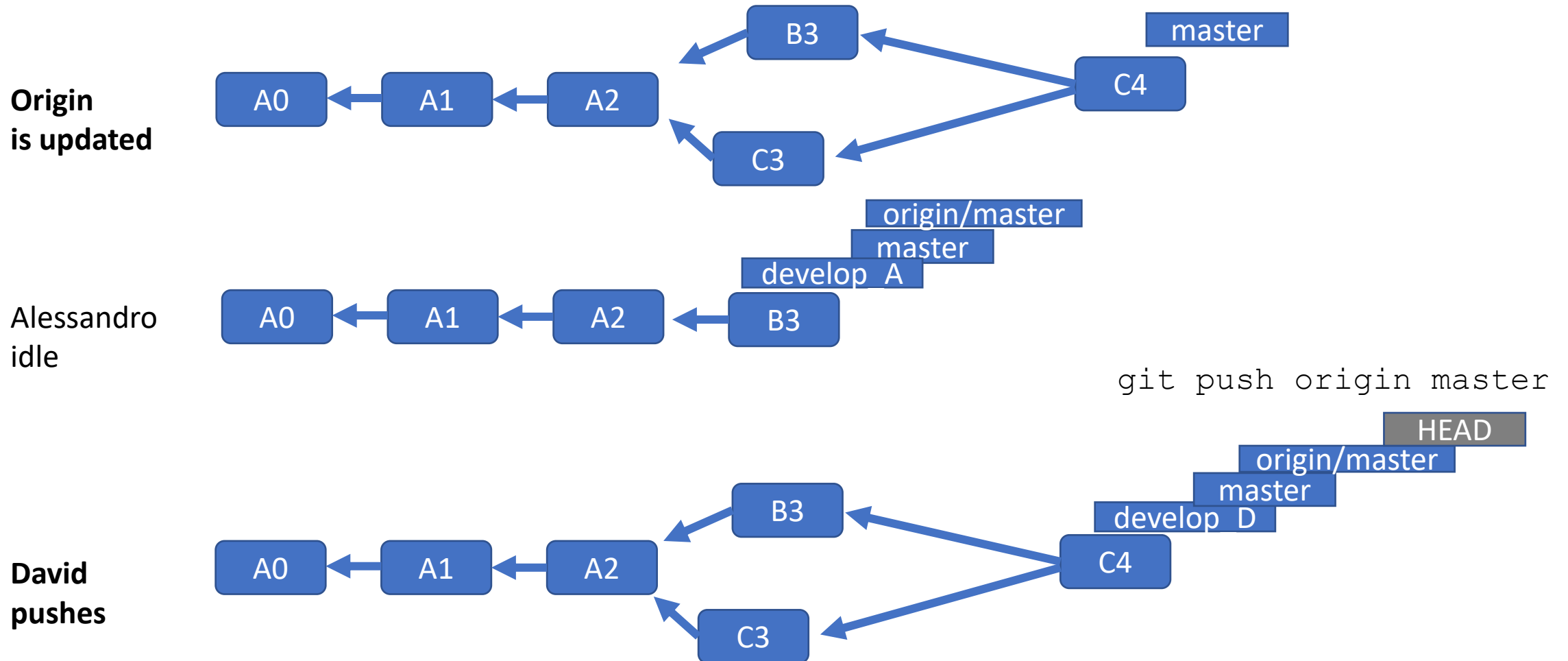
Collaborative development

What happens after clone?



Collaborative development

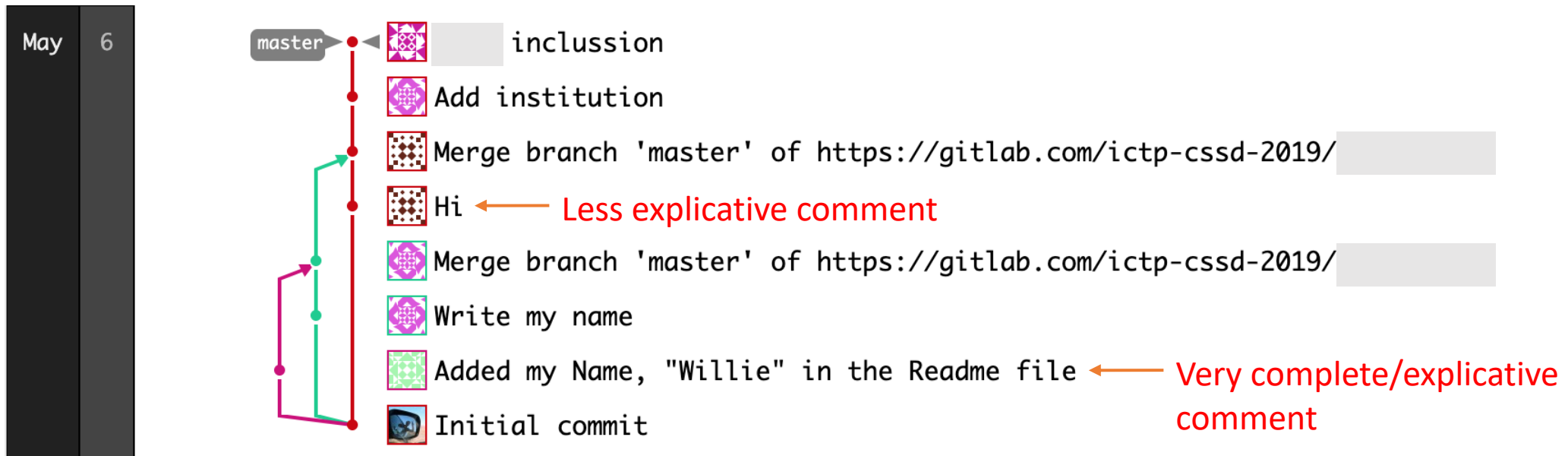
What happens after clone?



Maximizing scalability and effectiveness

- If all developers push to master, quality can degrade
- How to coordinate?
- How can we trust the changes?

- How to use this in research?



Maximizing scalability and effectiveness?

- If all developers push to master, quality can degrade
- How to coordinate?
- How can we trust the changes?

- How to use this in research?

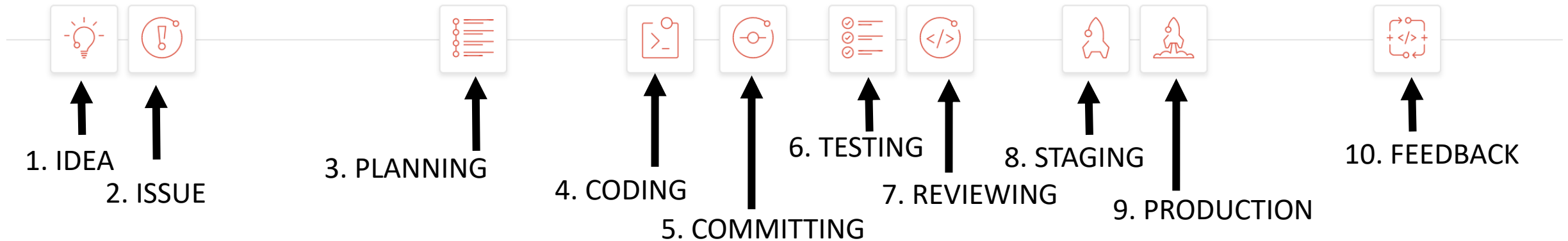


git



GitLab

The conversational development paradigm



Gitlab/github: meant to support this approach

Development model => content & conversations between developers

Fosters collaborations w/o centralized entities

In my experience: very scalable also in research (codes & execution)!

Super-structure to a Git repository

The screenshot shows the GitLab interface for the repository 'lbe-direct'. The main content area displays the repository details, including the name 'lbe-direct', the description 'The Lattice Boltzmann code under ftmake', and statistics for stars (3) and forks (0). Below this, there are links for 'Add Changelog', 'Add Contribution guide', and 'Add Kubernetes cluster'. The file browser shows a directory structure with folders like 'Auxiliary', 'Infrastructure', 'Tools', 'doc', 'http', 'performance', 'scripts', and 'src'. A merge request is visible, titled 'Merge branch 'develop-ISSUE283' into 'master'' by Federico Toschi, authored 2 weeks ago.

Annotations on the right side of the image explain the super-structure:

- Details**: What others developer did?
- Repository**: Content/commits
- Issues (132)**: Issues
- Merge Requests (0)**: Merging your contribution
- CI / CD**: Testing

Name	Last commit	Last update
Auxiliary	issue #211	3 years ago
Infrastructure	Added	2 years ago
Tools	[ISSUE #208] -- MM tools updates	5 months ago
doc	report Fluctuations update issue #259	4 years ago
http	Issue #181: sources for backend	4 years ago
performance	Resolved merge conflicts, thor cannot find H5. Tha...	4 years ago
scripts	[ISSUE #225] hacked problem possibly crashing te...	4 months ago
src	[ISSUE #283]	2 weeks ago
.cvsignore	*** empty log message ***	6 years ago

Issue at first

- I have an idea
- There is a bug
- I want to propose a new feature

The screenshot shows the GitLab interface for the 'kinectWebserver' project. The top navigation bar includes the GitLab logo, 'Projects', 'Groups', 'Activity', 'Milestones', 'Snippets', and a search bar. The left sidebar shows the project name 'kinectWebserver' and navigation options: Overview, Repository, Issues (28), List, Board, Labels, Milestones, and Merge Requests (0). The main content area displays the 'Issues' page for 'Alessandro C > kinectWebserver > Issues'. It features a filter bar with 'Open 28', 'Closed 86', and 'All 114'. A search bar is present with the text 'Search or filter results...'. Below the search bar, there are three issue entries:

- test failing** (#114) · opened a week ago by Alessandro C · updated a week ago
- Detections are made but not retained if tracking is turned off** (#113) · opened 3 months ago by Werner Kroneman · updated 3 months ago
- Integrate new cylinder-on-head tracking method into kinectWebserver** (#112) · opened 4 months ago by Werner Kroneman · updated 4 months ago

A fourth issue, 'Suspicious code in prune_and_undistort', is partially visible at the bottom. The 'New issue' button is highlighted with a red box.

Issue at first

- Case of a bug --> the issue got ticket number #113

The screenshot shows the GitLab interface for an issue. The top navigation bar includes the GitLab logo, project navigation (Projects, Groups, Activity, Milestones, Snippets), and search options. The left sidebar shows the project structure: kinectWebserver, Overview, Repository, Issues (28), Merge Requests (0), CI/CD, Registry, and Wiki. The main content area displays the issue details for #113, which is highlighted with a red box. The issue is titled "Detections are made but not retained if tracking is turned off" and is in an "Open" state, opened 3 months ago by Werner Kroneman. The description explains a bug where the detection step is carried out but the resulting JSON is empty because tracking is turned off. A code snippet is provided, showing a contradiction in the code's logic. The right sidebar shows various issue management options like Assignee, Milestone, Time tracking, Due date, Labels, Confidentiality, and Lock issue.

GitLab Projects Groups Activity Milestones Snippets This project Search 75 32

kinectWebserver Alessandro C > kinectWebserver > Issues #113

Open Opened 3 months ago by Werner Kroneman Close issue New issue

Detections are made but not retained if tracking is turned off

df_json is empty list if global_tracking is 0, despite the detection step being carried out and detection dots being visible on the merged image when global_detection is 1.

I suspect this code in pedestrian_vision is to blame:

```
## we need to produce this json just if there is no tracking. Otherwise it will be produced
if external_parameters.get_parameter("global_tracking",1) == 1:
    df_json = ret_merged['loc_df'].reset_index().to_dict(orient='records')
    print time_mark_pr.get_time_marked_string_prof("[ %s merge_and_show ] -
```

The comment contradicts the condition: it says it should only run if there is no tracking, yet the if-statement checks for a 1.

Not a high-priority issue on my side, though it was a bit confusing since I saw detection dots but no entries in the JSON.

Todo Add to

Assignee No assignee - assign yourself

Milestone None

Time tracking No estimate or time spent

Due date No due date

Labels None

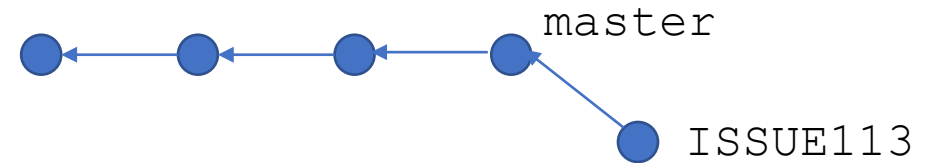
Confidentiality Not confidential

Lock issue

From issue to code

- If Ivan is in charge of fixing #113:

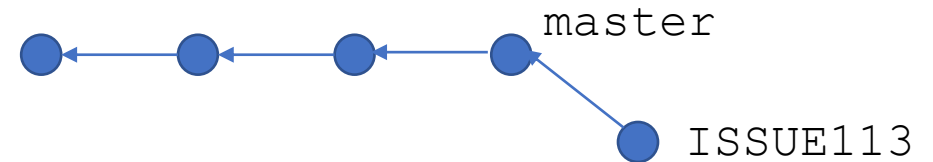
- [Clones the code – if he does not have already]
- Branches master -> e.g. into branch ISSUE113 (for trackability)
- `git checkout -b ISSUE113`



From issue to code

- If Ivan is in charge of fixing #113:

- [Clones the code – if he does not have already]
- Branches master -> e.g. into branch ISSUE113 (for trackability)
- `git checkout -b ISSUE113`

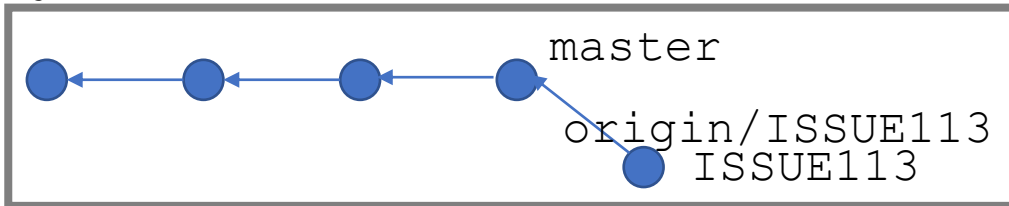


- Corrects bugs & make regression tests
- `git add code tests`
- `git commit -m '[ISSUE #113] - how code and tests have been changed'`
- `git push origin ISSUE113`

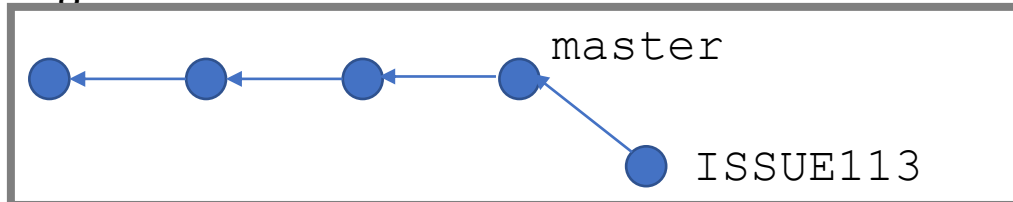
From issue to code

- If Ivan is in charge of fixing #113:
 - `git commit -m '[ISSUE #113] - how code and tests have been changed'`
 - `git push origin ISSUE113`

Ivan



Origin

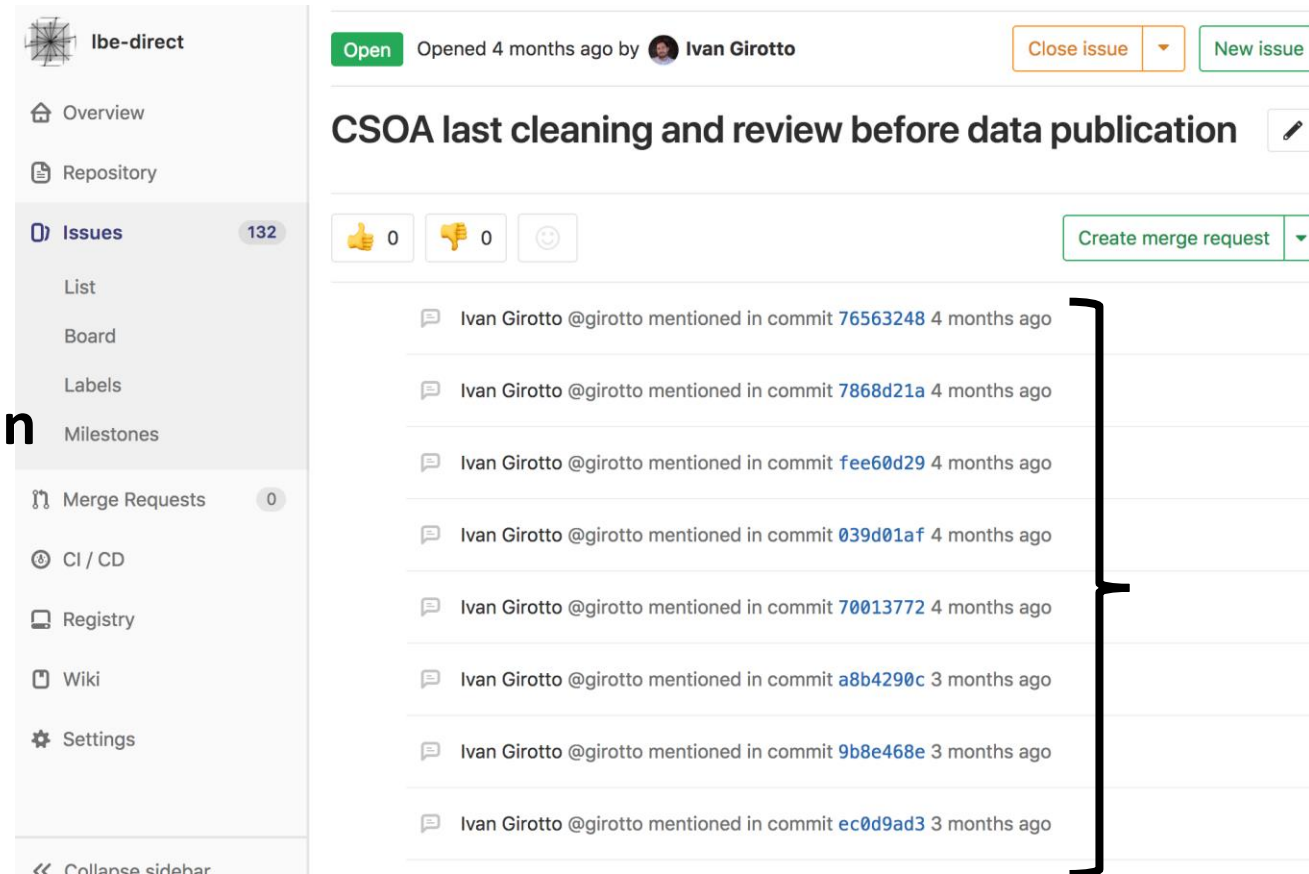


The screenshot shows the GitHub interface for the repository 'ibe-direct'. The left sidebar contains navigation links: Overview, Repository, Issues (132), List, Board, Labels, Milestones, Merge Requests (0), CI / CD, Registry, Wiki, and Settings. The main content area displays an issue titled 'CSOA last cleaning and review before data publication', opened 4 months ago by Ivan Girotto. The issue is currently open and has 0 thumbs up and 0 thumbs down. A 'Create merge request' button is visible. The issue history shows several mentions by Ivan Girotto in various commits, including 76563248, 7868d21a, fee60d29, 039d01af, 70013772, a8b4290c, 9b8e468e, and ec0d9ad3.

From issue to code

- If Ivan is in charge of fixing #113:
 - `git commit -m '[ISSUE #113] - how code and tests have been changed'`
 - `git push origin ISSUE113`

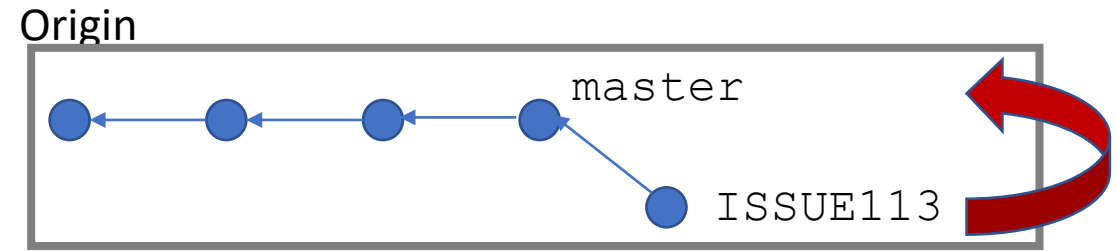
ALWAYS DO specify the issue number in the commit message. This will **link** each **contribution** to the related **motivation**



The screenshot shows the GitHub interface for the repository 'lbe-direct'. The left sidebar contains navigation options: Overview, Repository, Issues (132), List, Board, Labels, Milestones, Merge Requests (0), CI / CD, Registry, Wiki, and Settings. The main content area displays an issue titled 'CSOA last cleaning and review before data publication' opened 4 months ago by Ivan Girotto. Below the issue title are thumbs up (0) and thumbs down (0) buttons, and a 'Create merge request' button. A list of commit messages is shown, each mentioning the issue: 'Ivan Girotto @girotto mentioned in commit 76563248 4 months ago', 'Ivan Girotto @girotto mentioned in commit 7868d21a 4 months ago', 'Ivan Girotto @girotto mentioned in commit fee60d29 4 months ago', 'Ivan Girotto @girotto mentioned in commit 039d01af 4 months ago', 'Ivan Girotto @girotto mentioned in commit 70013772 4 months ago', 'Ivan Girotto @girotto mentioned in commit a8b4290c 3 months ago', 'Ivan Girotto @girotto mentioned in commit 9b8e468e 3 months ago', and 'Ivan Girotto @girotto mentioned in commit ec0d9ad3 3 months ago'. A large black bracket on the right side of the commit list groups these messages together.

Code review merge request

```
git push origin ISSUE113
```



master is a privileged branch. Developers must not push directly! (generally is also prohibited)

- **Code review from authoritative parties is required**
- **Ask the authoritative part to merge your contribution through a merge request**

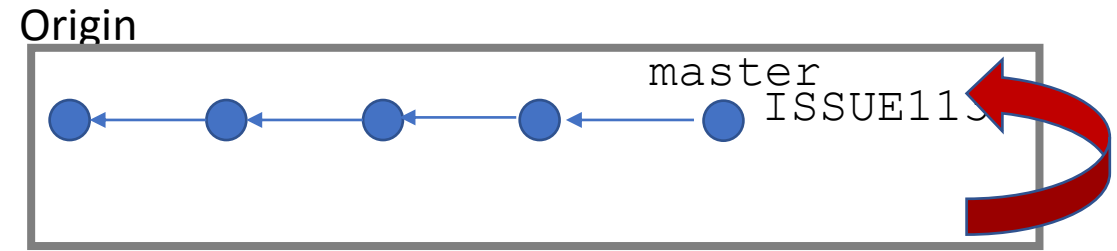
codes > lbe-direct > Merge Requests

Open 0 Merged 87 Closed 35 All 122

Edit merge requests **New merge request**

Search or filter results... Created date

Code review merge request



GitLab Projects Groups Activity Milestones Snippets This project Search 75 32

ftmake

- Overview
- Repository
- Issues 29
- Merge Requests 0**
- CI / CD
- Wiki
- Settings

Merged Opened a week ago by Gianluca Di Staso Edit Close merge request

Issue 102

Modification to make the `dsmc_indirect` code run under ftmake.

Request to merge ISSUE_102 into master

- ✓ Pipeline #13356 passed for e99181e7. ✓ ✓
- ✓ Merged by Federico Toschi 2 days ago Revert Cherry-pick

The changes were merged into **master**

You can remove source branch now Remove Source Branch

👍 0 👎 0 😊

Discussion 0 Commits 4 Pipelines 4 Changes 4

- Gianluca Di Staso @gianluca added 1 commit 4 days ago
 - e99181e7 - [ISSUE #102] -- Add new names of cartesianus

Compare with previous version

Todo Add todo >>

Assignee Edit
Federico Toschi @toschi

Milestone Edit
None

Time tracking ⓘ
No estimate or time spent

Labels Edit
None

Lock merge request Edit
Unlocked

2 participants

Notifications

Reference: toschi/ftmake!71

Code review merge request

Code can (should) be reviewed
before merging by an authority

Discussion 0 Commits 4 Pipelines 4 Changes 4

Changes between latest version and master

Showing 4 changed files with 22 additions and 1 deletion

Hide whitespace changes Inline Side-by-side

▼ CMakeLists.txt

Line	Left (latest version)	Right (master)
182	#####	#####
183	set(CMAKE_MODULE_PATH \${CMAKE_MODULE_PATH} "\${FTMAKE_SOURCE_DIR}/cmake")	set(CMAKE_MODULE_PATH \${CMAKE_MODULE_PATH} "\${FTMAKE_SOURCE_DIR}/cmake")
184	set(CMAKE_MODULE_PATH \${CMAKE_MODULE_PATH} "\${PROJECT_SOURCE_DIR}/cmake")	set(CMAKE_MODULE_PATH \${CMAKE_MODULE_PATH} "\${PROJECT_SOURCE_DIR}/cmake")
185	- #message(STATUS \${PROJECT_SOURCE_DIR})	+ message(STATUS \${PROJECT_SOURCE_DIR})
186	#####	#####
187	#####	#####
188	# Here I invoke the loading of machine specific hooks	# Here I invoke the loading of machine specific hooks
769	SET_TARGET_PROPERTIES(\${TARGET_NAME} PROPERTIES LINKER_LANGUAGE Fortran)	SET_TARGET_PROPERTIES(\${TARGET_NAME} PROPERTIES LINKER_LANGUAGE Fortran)
770	endif(EXISTS "\${PROJECT_SOURCE_DIR}/main.F90")	endif(EXISTS "\${PROJECT_SOURCE_DIR}/main.F90")

Assignee: Federico Toschi (@toschi)

Milestone: None

Time tracking: No estimate or time spent

Labels: None

Lock merge request: Unlocked

2 participants

Notifications

Reference: toschiftr

Nightly testing

- Did the development of the day worked ok?
 - 11PM – start test suit current dev branch
 - 8AM – see problems; fix
- Issues?

Aims

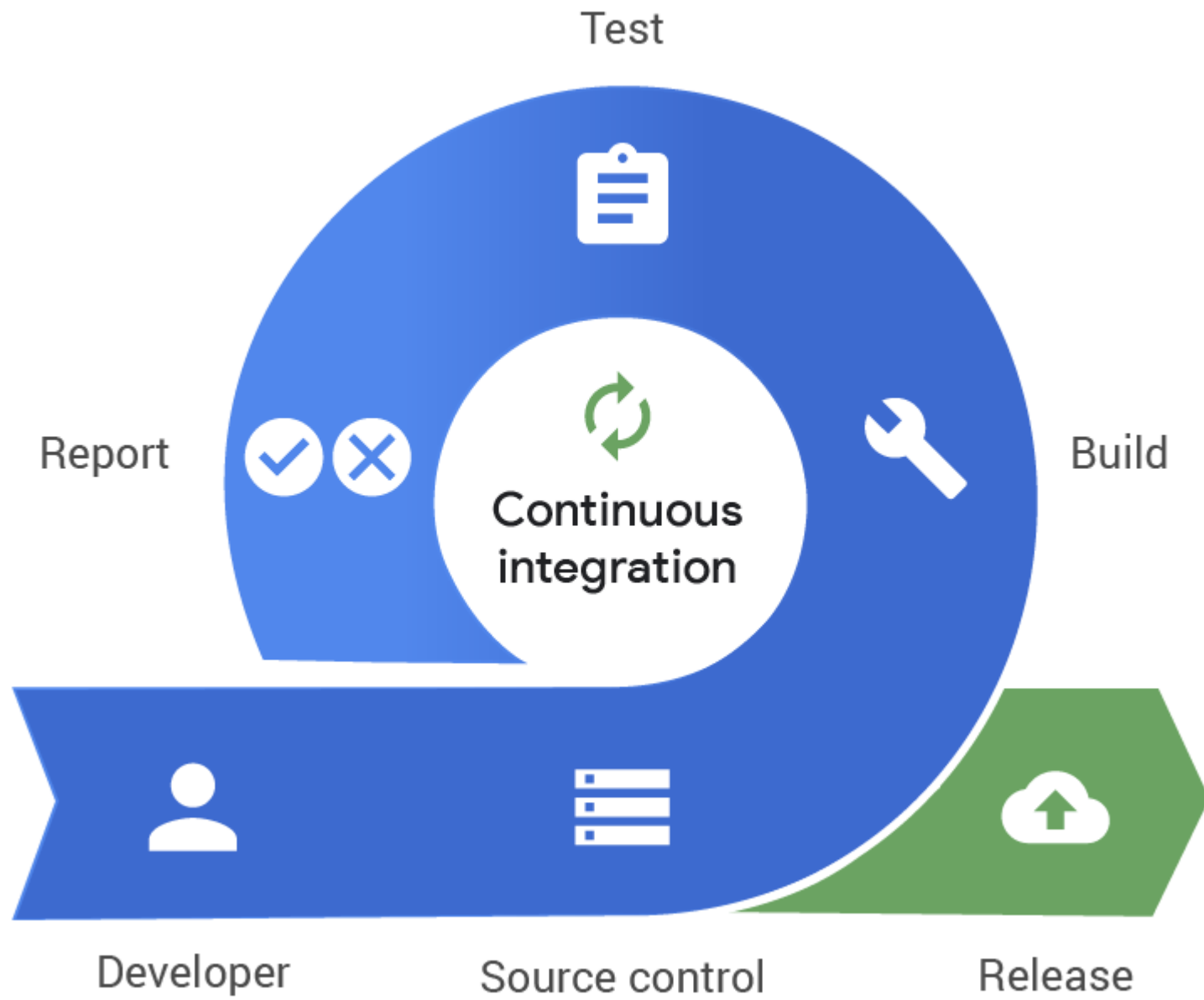
- Understand quickly if code (each branch) pass tests
- Merge frequently; avoid drifts
- Test before merging and merge
- Tests run quickly ($O(10\text{min})$)

Also

- Deploy frequently – “agile” dev-user cycle

Coding: collaborative,
trustworthy, reproducible

Alessandro Corbetta



Automated testing

- We want each commit tested (“by trusted party”)
- We want to merge codes only if they **pass tests!**
- **Continuous integration**
each commit that we push to the origin is tested

Request to merge [ISSUE_102](#) into [master](#)

✓ Pipeline #13356 passed for [e99181e7](#). ✓ ✓

✓ Merged by [Federico Toschi](#) 2 days ago [Revert](#) [Cherry-pick](#)

The changes were merged into [master](#)

You can remove source branch now [Remove Source Branch](#)

16 Apr, 2018 2 commits

Merge br
Federico Toschi authored 2 weeks ago ✓ 25285389

[ISSUE #283] 3D
Ivan Girotto authored 2 weeks ago 035f96ea

04 Apr, 2018 1 commit

Merge branch
Federico Toschi authored a month ago ✓ a5167bdf

03 Apr, 2018 2 commits

[ISSUE #279] hacke
Ivan Girotto authored a month ago 97be4f06

[ISSUE #279] Wor
Ivan Girotto authored a month ago 332d16bb

20 Mar, 2018 3 commits

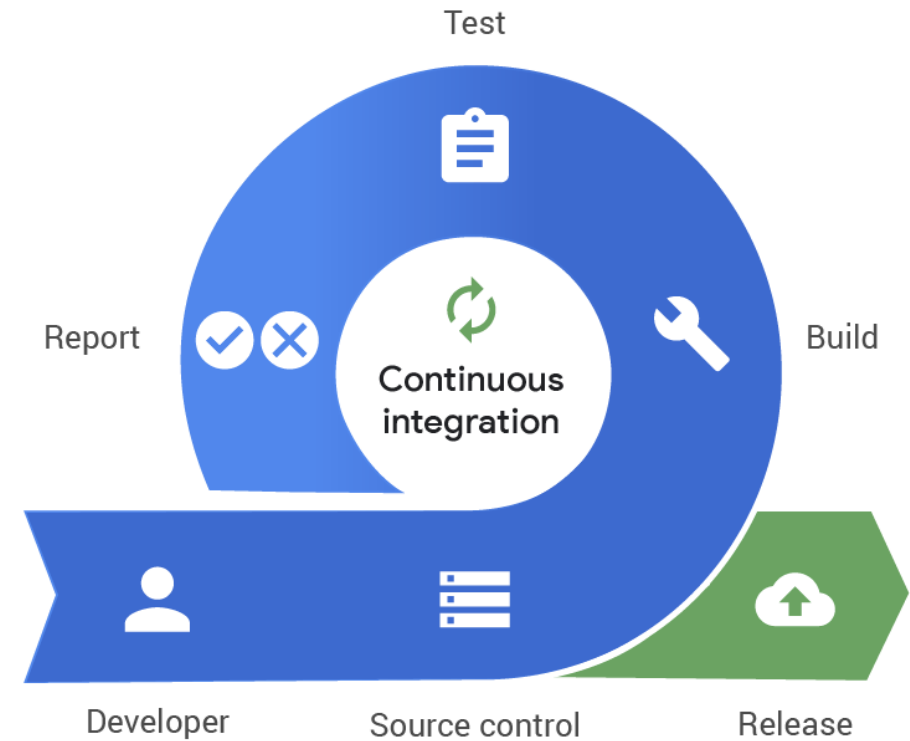
Merg
Federico Toschi authored a month ago ✓ 85de5d29

Merg
Federico Toschi authored a month ago ✓ 8e0f1dc0

I can merge safely: tests are passed

Continuous integration

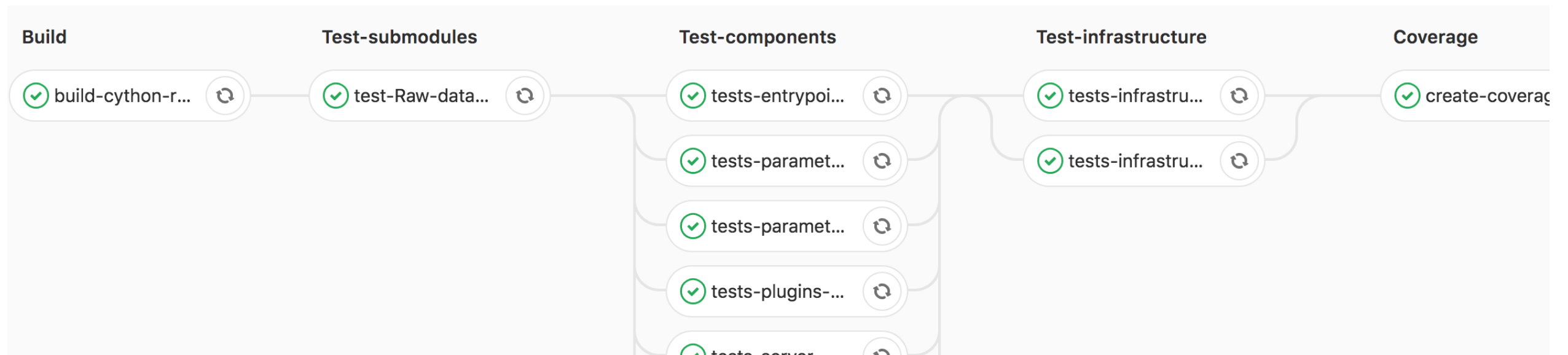
- **Each push: automated** remote **testing**
- Every user fully aware of the code state




- If testing quick, dev cycle & master merge: very frequent
 - Many github repos: hundreds merge per day after remote testing

Continuous integration in gitlab

- a file in the repo root
`.gitlab-ci.yml`
tells the server how to run tests and in which environment
- Can be one **single test** case or a very complicated testing pipeline with cross-test dependence.



Minimal python example

 my_code.py  159 bytes

```
1 def summing_f(a,b):
2     print(a,b)
3     return a+b
4
5 def multiplying_f(a,b):
6     print(a,b)
7     return a*b
8
9
10 if __name__ == '__main__':
11     print(summing_f(2,3))
12
```

 test_code.py  141 bytes

```
1 from my_code import summing_f
2
3
4 def test_summing_f1():
5     assert summing_f(1,1) == 2
6
7 def test_summing_f2():
8     assert summing_f(4,1) == 5
9
```

Minimal python example: CI

`.gitlab-ci.yml`

```
image: python:latest
```

```
test-only:
```

```
  script:
```

```
    - pip install pytest pytest-cov
```

```
    - pytest -vvv
```


Minimal python example: CI

`.gitlab-ci.yml`

```
image: python:latest
```



Running in a linux sandbox with up-to-date python (Docker container)

```
test-only:
```

```
  script:
```

Installing dependencies, here put numpy etc...
(better with requirements.txt)

```
    - pip install pytest
```



```
    - pytest -vvv
```



Testing



Search GitLab

passed

Job test-only triggered 5 minutes ago by Administrator



test-only
test



Pytest Example

Project information

Repository

Issues 0

Merge requests 0

CI/CD

Pipelines

Editor

Jobs

Schedules

Security & Compliance

Deployments

Packages and registries

Infrastructure

Monitor

Analytics

Wiki

Snippets

Settings

```
1 Running with gitlab-runner 15.6.1 (133d7e76)
2   on Ci runner NS9smqSy
3 Preparing the "docker" executor
4 Using Docker executor with image python:latest ...
5 Pulling docker image python:latest ...
6 Using docker image sha256:ee4e7a0f1c354d9996229a765d0785df2671252c1822ae111015d37dcf5f765b for
7
8 Preparing environment
9 Running on runner-ns9smqsy-project-2-concurrent-0 via smr3696-1...
10
11 Getting source from Git repository
12 Fetching changes with git depth set to 20...
13 Reinitialized existing Git repository in /builds/gitlab-instance-59615060/pytest-example/.git/
14 Checking out 2733dd46 as main...
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38 $ pytest -vvv
39 ===== test session starts =====
40 platform linux -- Python 3.11.0, pytest-7.2.0, pluggy-1.0.0 -- /usr/local/bin/python
41 cachedir: .pytest_cache
42 rootdir: /builds/gitlab-instance-59615060/pytest-example
43 plugins: cov-4.0.0
44 collecting ... collected 2 items
45
46 test_code.py::test_summing_f1 PASSED [ 50%]
47
48 test_code.py::test_summing_f2 PASSED [100%]
49
50 ===== 2 passed in 0.01s =====
51
52 Cleaning up project directory and file based variables
53
54 Job succeeded
```

See examples

[https://git.smr3696.ictp.it/gitlab-instance-59615060/**pytest-example**](https://git.smr3696.ictp.it/gitlab-instance-59615060/pytest-example)

Continuous deployment

- One step further: serving the product immediately
- Example: static pages that serve **docs, executables...**

Best practices

- **Edits always follow issues**
- Pull/commit/push/merge request often.
 - You don't want your version to diverge
- Make meaningful commit message
- **Commit message** include ticket number
- **NEVER commit products, just source.**
 - (products can be made in cloud by continuous integration)
 - E.g. documentation can be generated by CI
- **Never commit large binary files.** Git does not properly understand those (git-lfs)

Minmal python example: testing

```
image: python:latest

variables:
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache"

cache:
  paths:
    - .cache/pip
    - venv/

before_script:
  - python -V          # Print out python version for debugging
  - pip install virtualenv
  - virtualenv venv
  - source venv/bin/activate
  - pip install numpy nose

test:
  script:
    - cd binary_str_2_float
    - nosetests -v
```

`.gitlab-ci.yml`

Building continuous integration pipelines

Case of our python exercise

`.gitlab-ci.yml`

image: python:latest

variables:

```
PIP_CACHE_DIR: "${CI_PROJECT_DIR}/.cache"
```

cache:

paths:

- .cache/pip
- venv/

before_script:

- python -V # Print out python version for debugging
- pip install virtualenv
- virtualenv venv
- source venv/bin/activate
- pip install numpy pytest pytest-cov

test:

script:

- cd binary_str_2_float
- pytest -v

We will run the tests in a “virtual linux machine” which runs the latest python version. (Docker container)

Building continuous integration pipelines

Case of our python exercise

```
image: python:latest
```

```
.gitlab-ci.yml
```

```
variables:
```

```
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache"
```

```
cache:
```

```
  paths:
```

- .cache/pip
- venv/

```
before_script:
```

- python -V # Print out python version for debugging
- pip install virtualenv
- virtualenv venv
- source venv/bin/activate
- **pip install numpy pytest pytest-cov**

The machine is empty. We need to configure it from scratch every time. (good for reproducibility)/

```
test:
```

```
  script:
```

- cd binary_str_2_float
- pytest -v

Building continuous integration pipelines

Case of our python exercise

```
image: python:latest
```

```
variables:
```

```
  PIP_CACHE_DIR: "$CI_PROJECT_DIR/.cache"
```

```
cache:
```

```
  paths:
```

- .cache/pip
- venv/

```
before_script:
```

- python -V # Print out python version for debugging
- pip install virtualenv
- virtualenv venv
- source venv/bin/activate
- pip install **numpy pytest pytest-cov**

```
test:
```

```
script:
```

- **cd binary_str_2_float**
- **pytest -v**

.gitlab-ci.yml

Calls the tests and captures the value fail/pass return value

Result is sent back to the server.