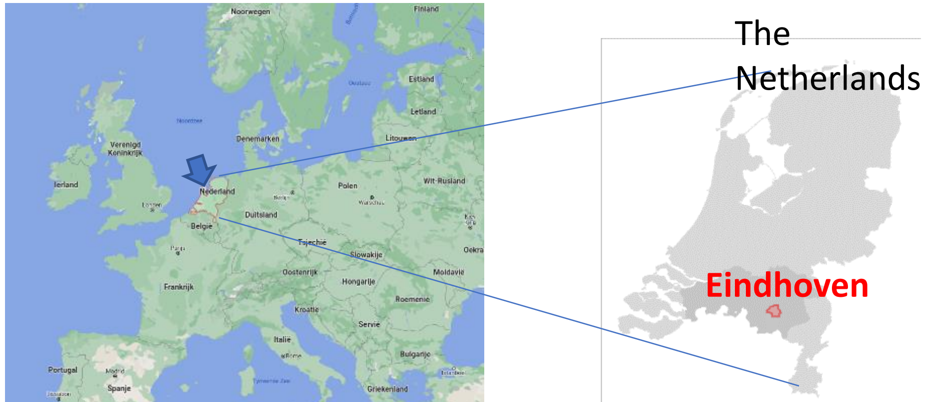
Alessandro Corbetta

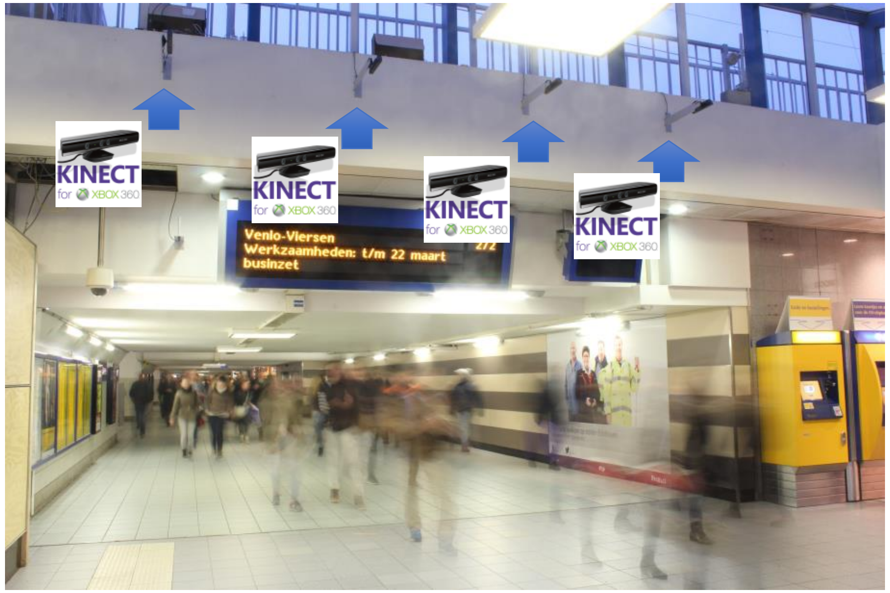
- **Pedestrian dynamics & traffic as active fluid**
- **Machine Learning for fluid mechanics**

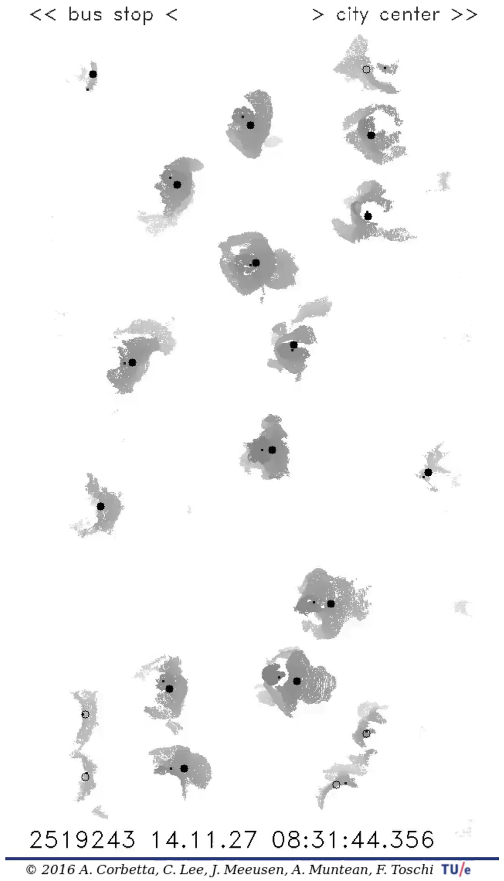


TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Department of Applied Physics
Fluids and Flows

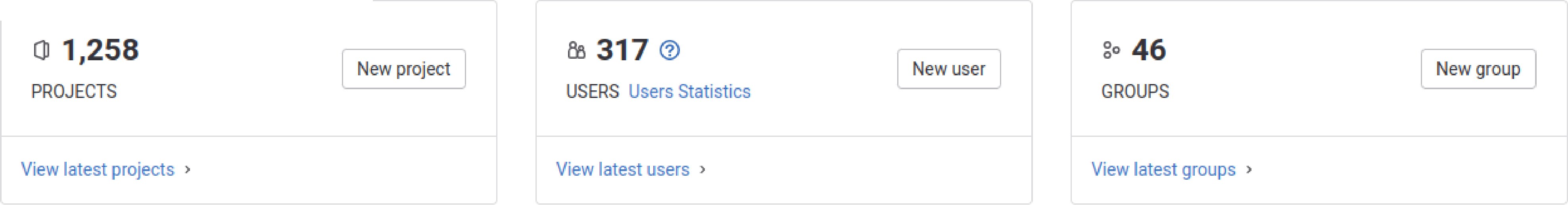
https://crowdflow.phys.tue.nl




<< bus stop <
> city center >>
2519243 14.11.27 08:31:44.356
© 2016 A. Corbetta, C. Lee, J. Meeusen, A. Muntean, F. Toschi TU/e

Alessandro Corbetta

GitLab Since 2015



1,258 PROJECTS

317 USERS

46 GROUPS

QLUSTAR

Computing cluster

PROXMOX

Virtualization cluster, Hyperconverged storage

Coding: collaborative, trustworthy, reproducible

Alessandro Corbetta

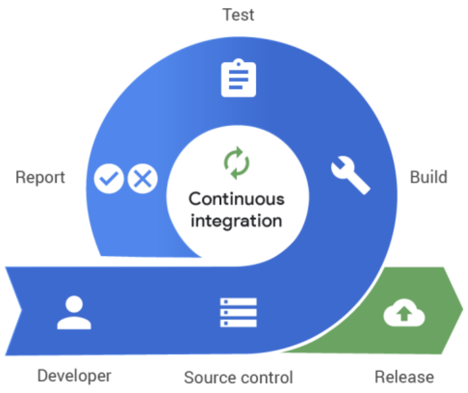
Plan of the lectures

Structured collaboration

Gitlab
Conversational development
DevOps

Trusting changes

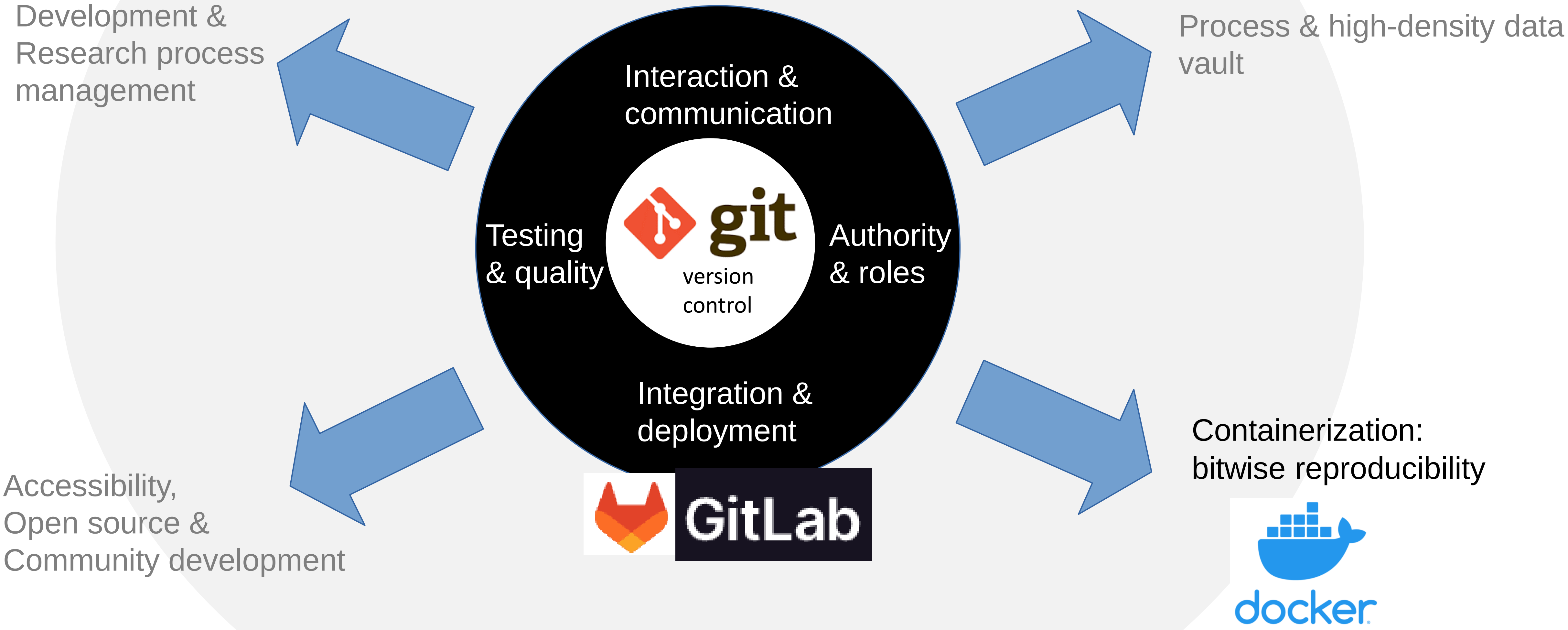
Testing (Ali)
Continuous integration
Continuous deployment



Immediate reproducibility

Containers
Docker
Virtualenvs

An infrastructure in shells



Development & Research process management

Process & high-density data vault

Interaction & communication

Testing & quality

version control

Authority & roles

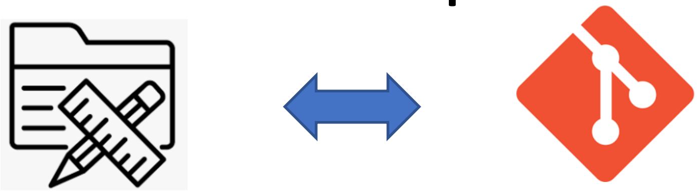
Integration & deployment

Accessibility, Open source & Community development

Containerization: bitwise reproducibility

Version control with 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

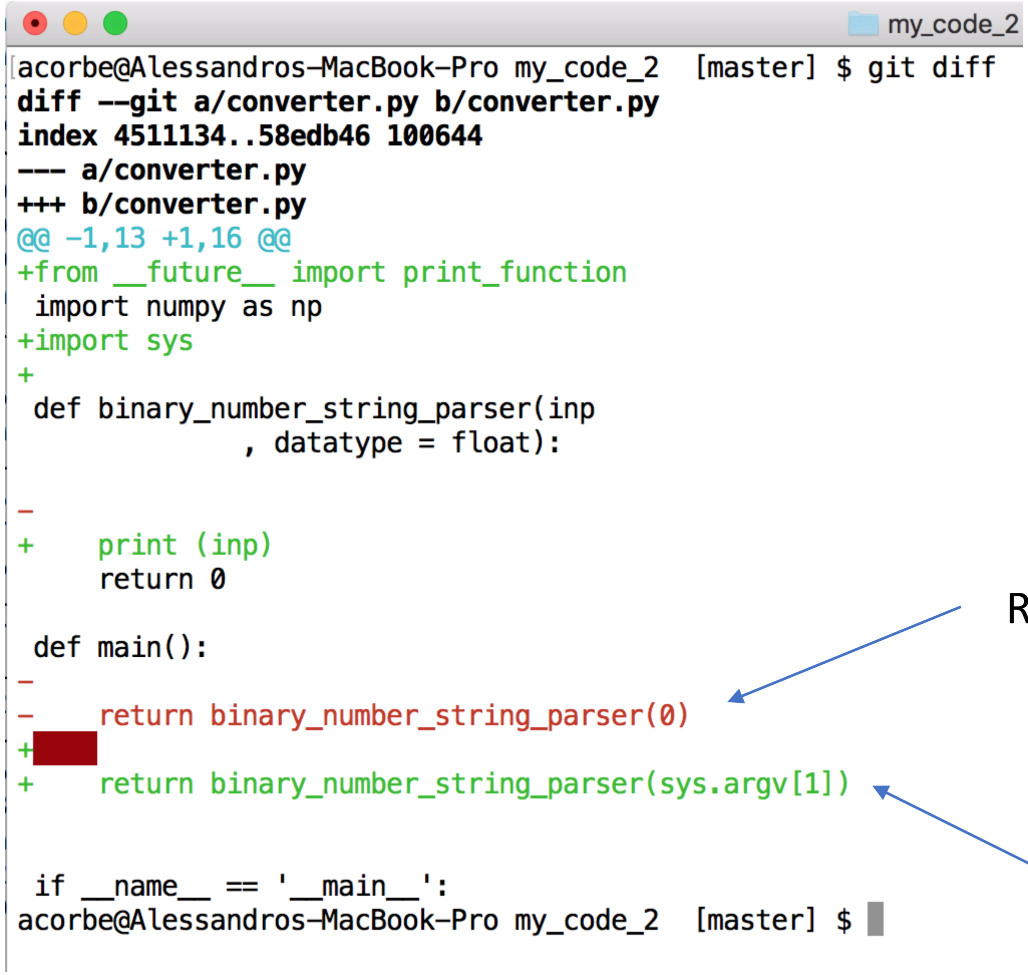
Make a snapshot of the current version

```
$ 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`

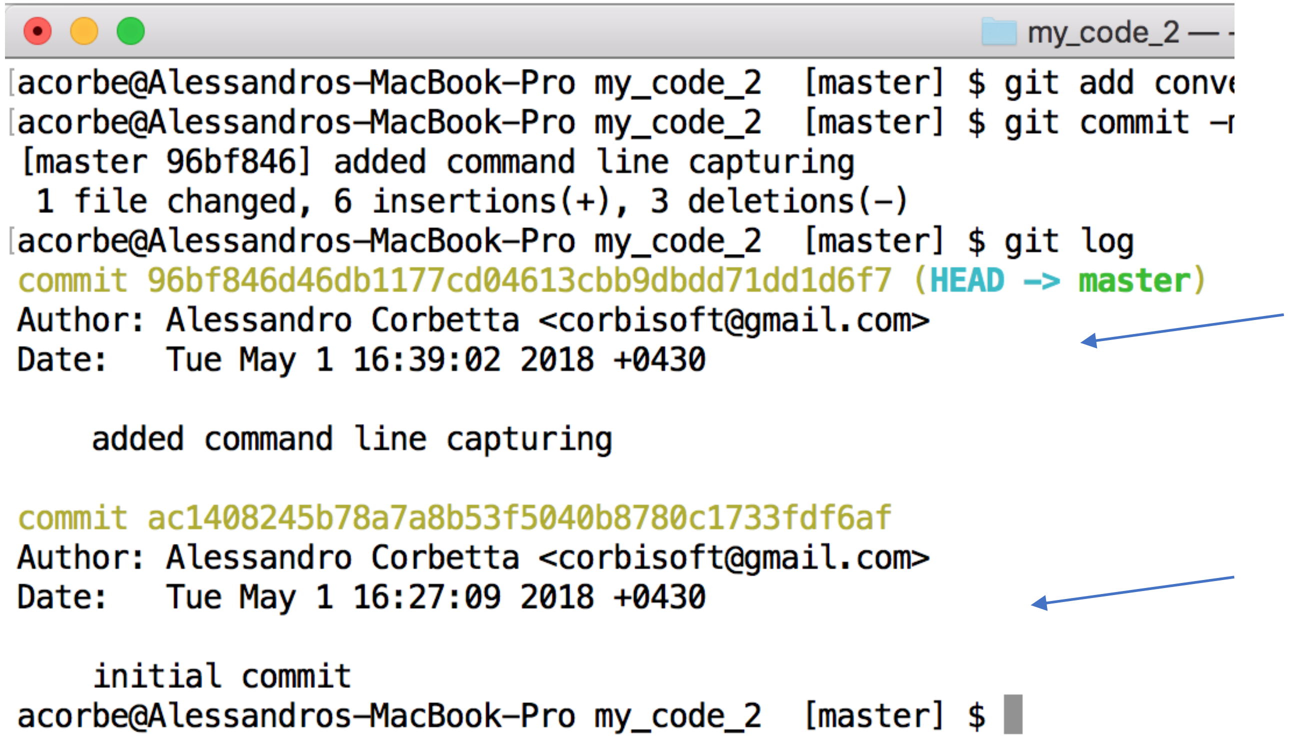


```
acorbe@Alessandros-MacBook-Pro my_code_2  [master] $ git diff
diff --git a/converter.py b/converter.py
index 4511134..58edb46 100644
--- a/converter.py
+++ b/converter.py
@@ -1,13 +1,16 @@
+from __future__ import print_function
 import numpy as np
+import sys
+
 def binary_number_string_parser(inp
              , 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__':
acorbe@Alessandros-MacBook-Pro my_code_2  [master] $
```

Removed content

Added content

Changes
`git log`



```
[acorbe@Alessandros-MacBook-Pro my_code_2  [master] $ git add conve
[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] $
```

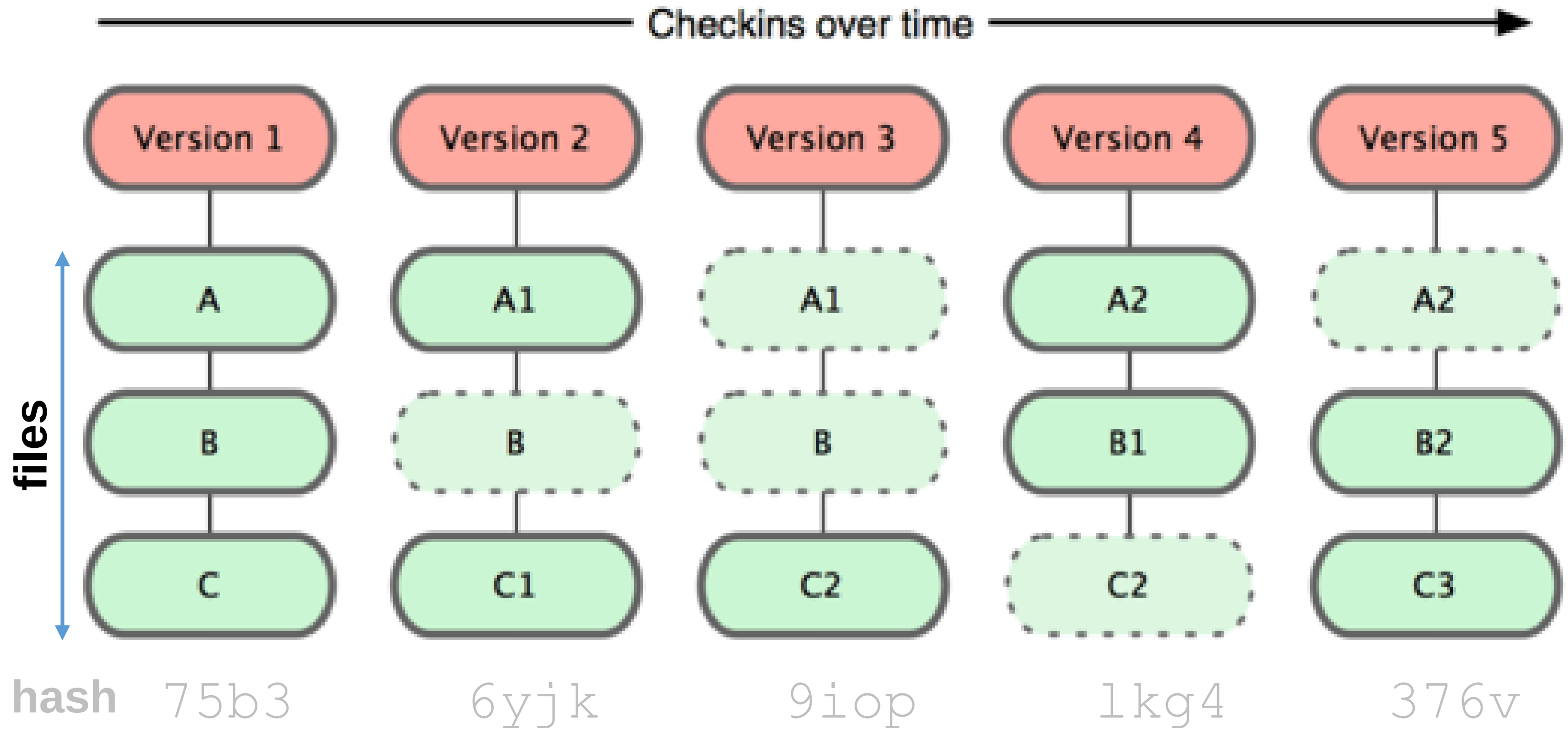
Second snapshot (commit)

First snapshot (commit)

History & rollback

- Each commit holds a complete snapshot of the repository

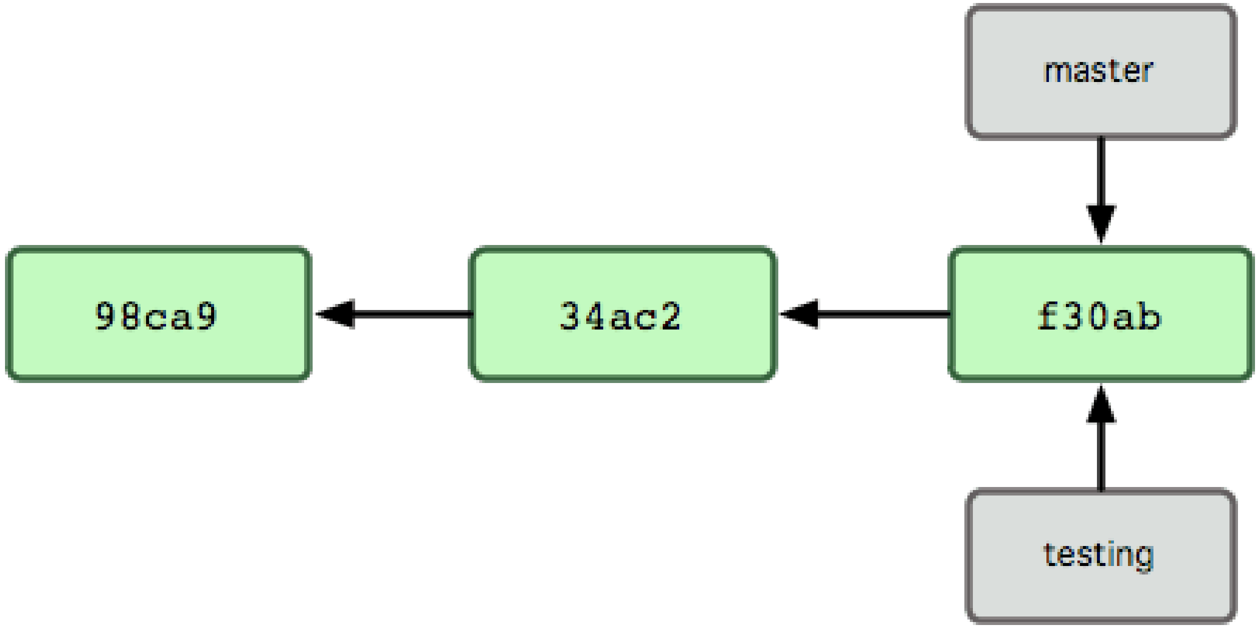
```
git checkout <hash>
```



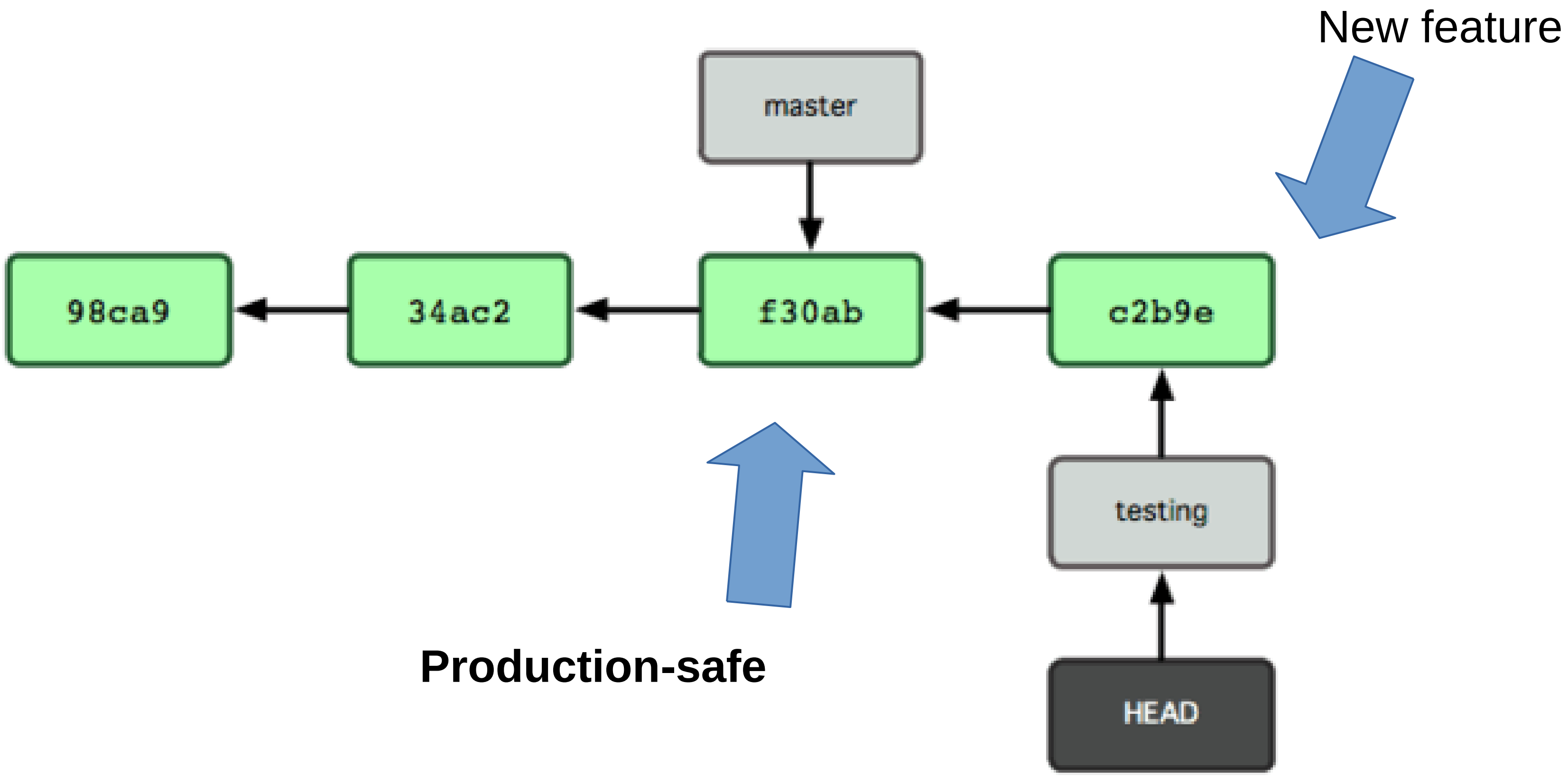
Branches for a safe future

Default branch “`master`” or “`main`”

New branch: `git **branch** testing`



Development in “testing”

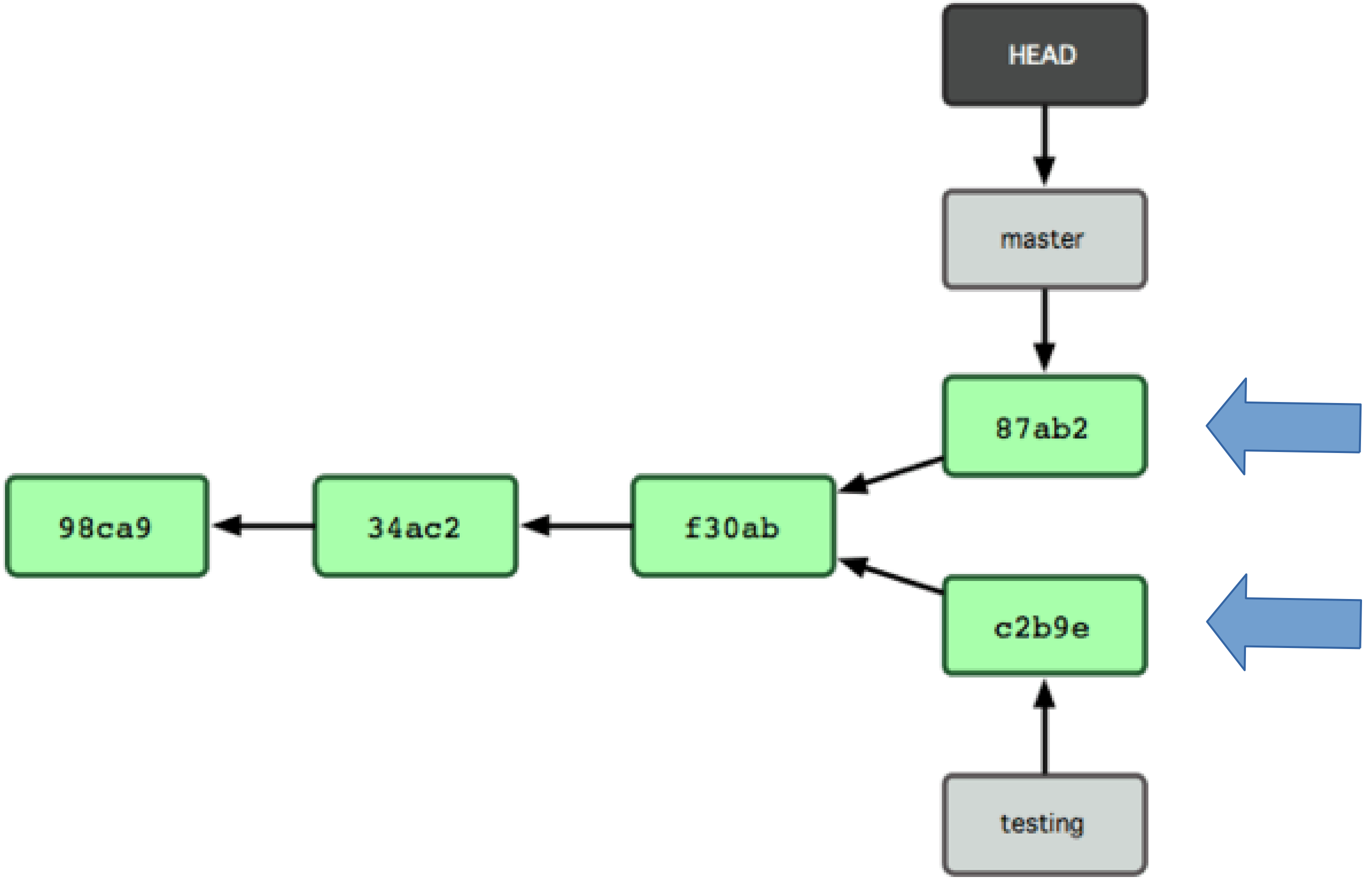


New feature

Production-safe

Branches can evolve independently!

(typically is not what you wanna do, master is only one!)

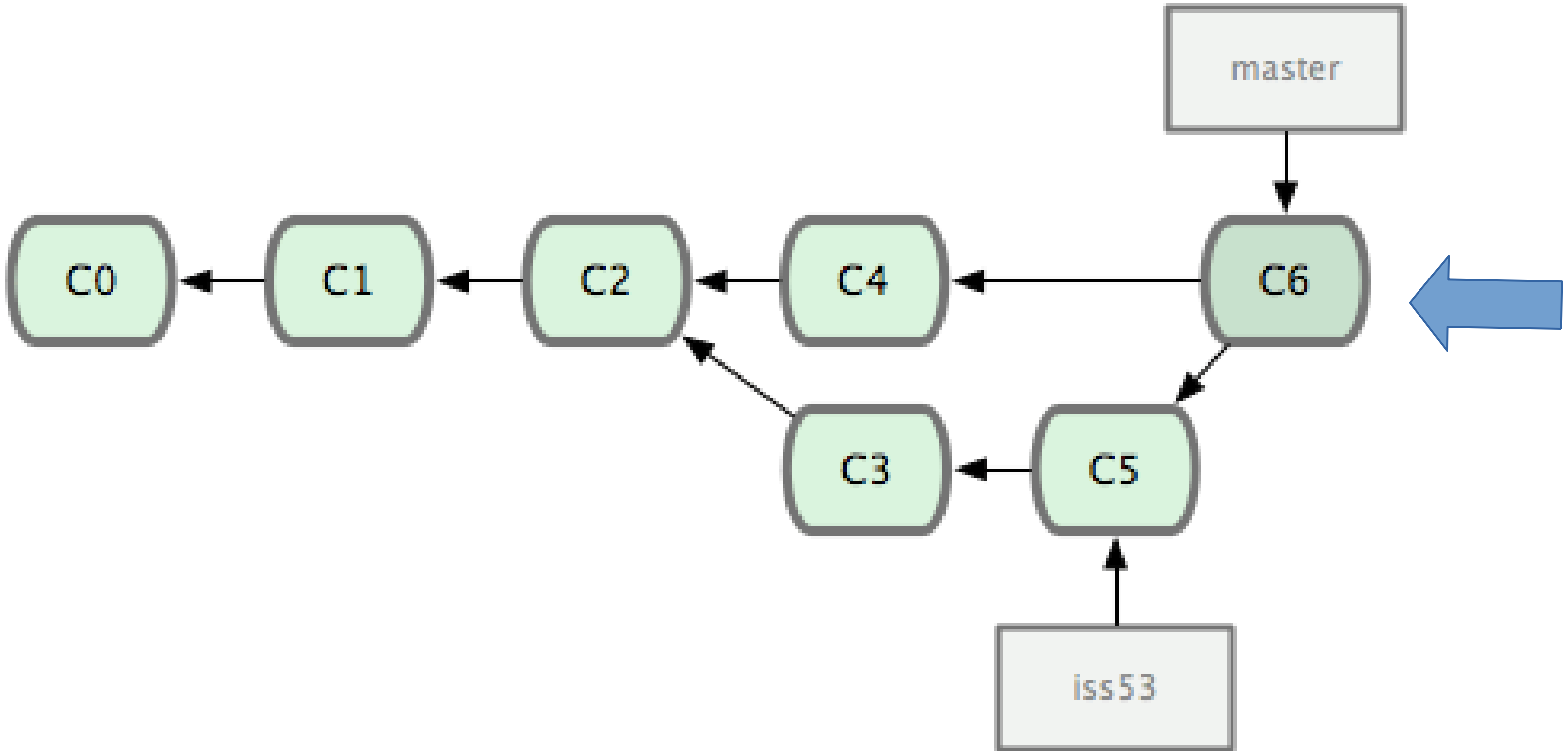


Production

Development

Merging (to the the only master ☺)

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

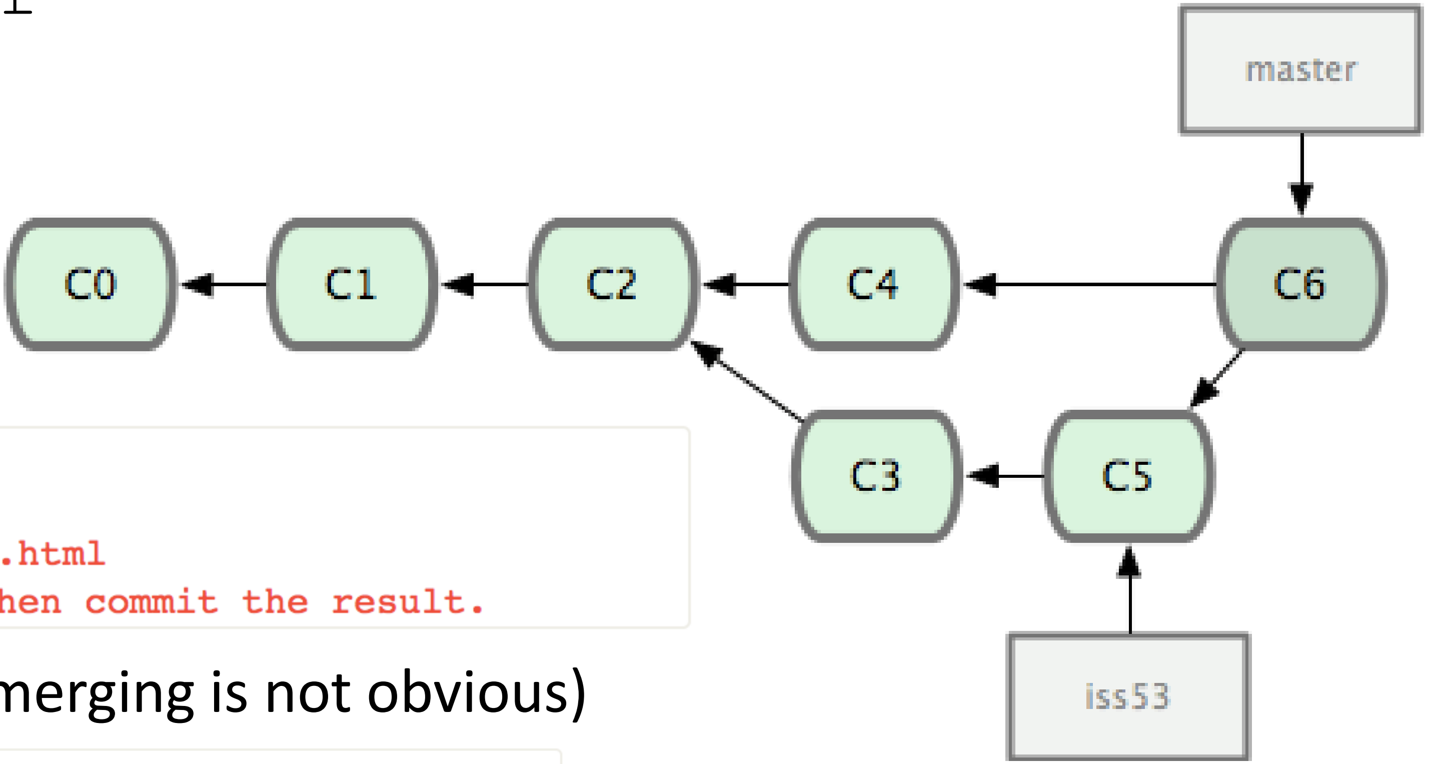


Feature iss53 is brought
To production

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

(1 developer)

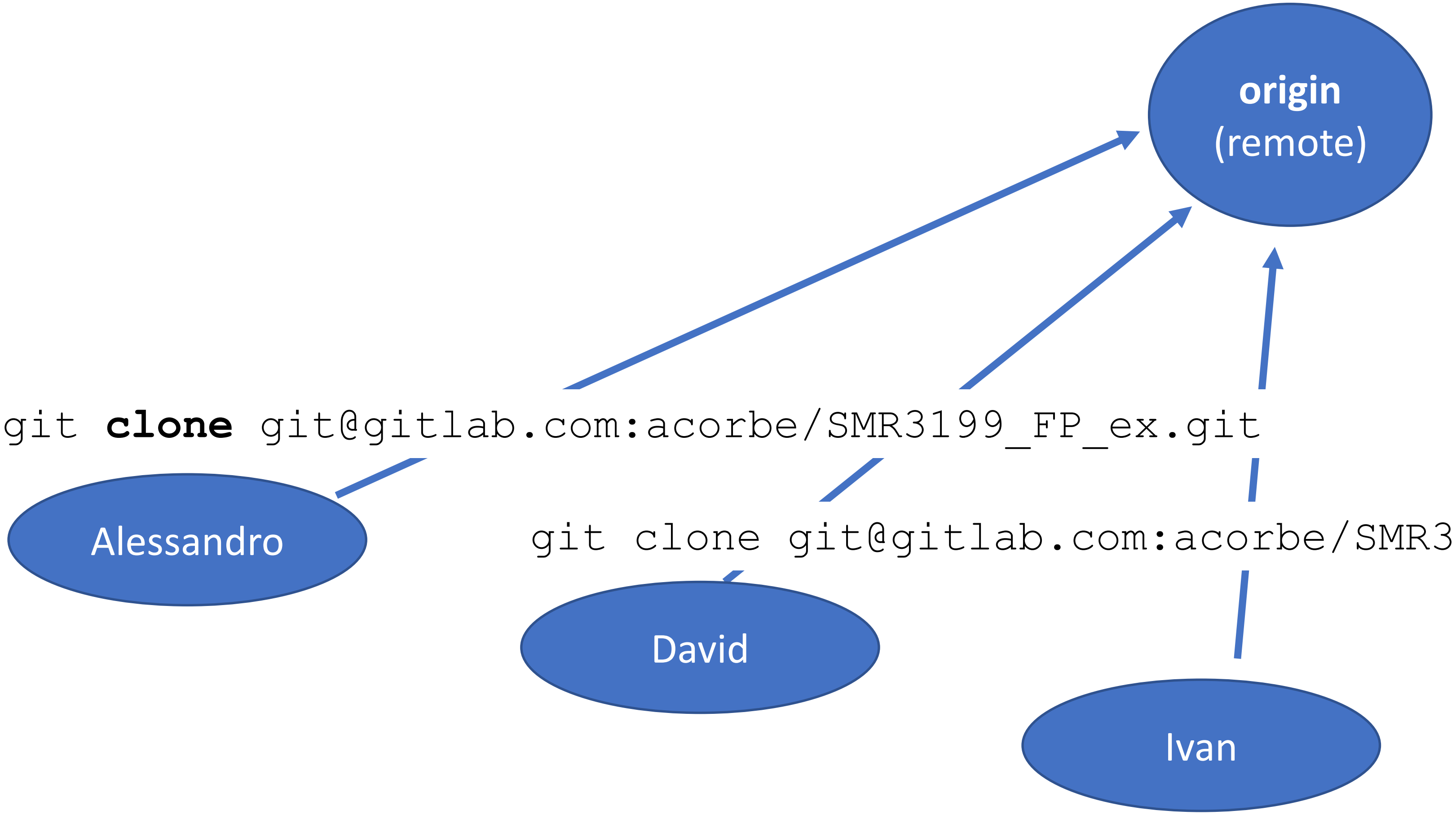
• git init	
• git add	stage file for snapshot
• git commit	take snapshot
• git diff	compare snapshots
• git log	snapshot list
• git checkout	move to a different snapshot
• git branch	branch the future
• git merge	join different futures in one

Coding: **collaborative**, trustworthy, reproducible

Alessandro Corbetta

Collaborative development

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



```
git clone git@gitlab.com:acorbe/SMR3199_FP_ex.git
```

```
git clone git@gitlab.com:acorbe/SMR3199_FP_ex.git
```

```
git clone git@gitlab.com:acorbe/SMR3199_FP_ex.git
```

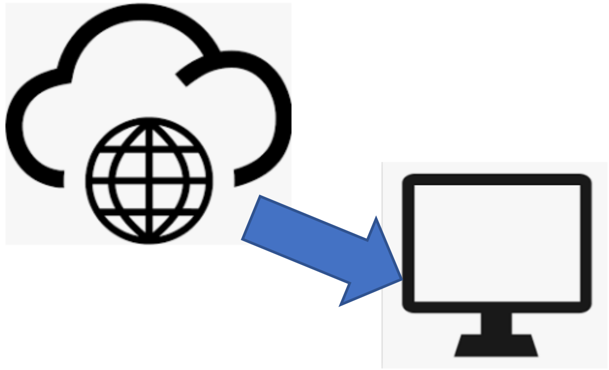
GitLab

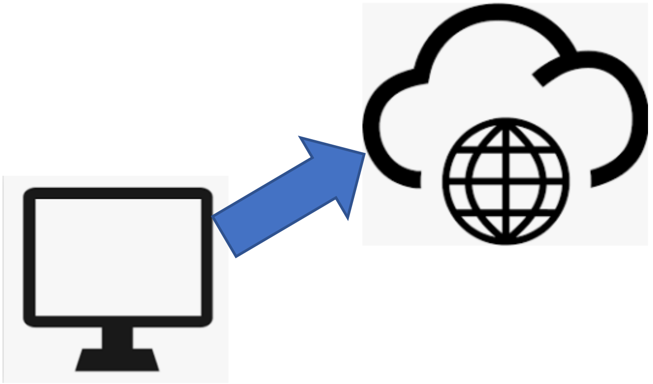
GitHub

Bitbucket

Multiple solutions to host the server yourself

Pulling/Pushing commits to the origin





To update <branch> locally

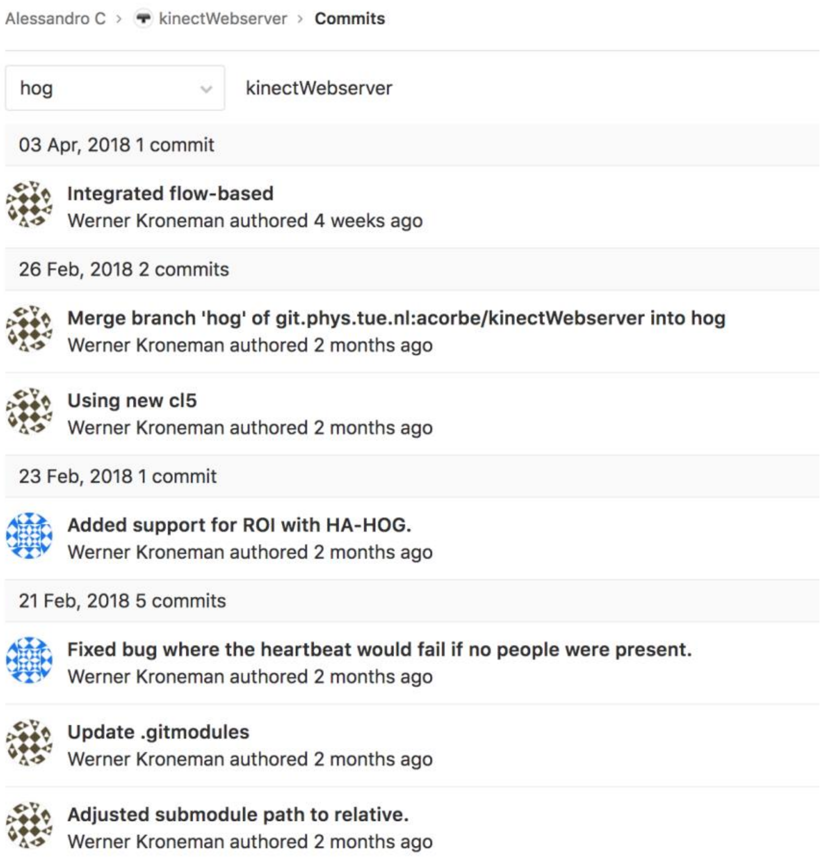
- git checkout <branch>
- git **pull** origin <branch>

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

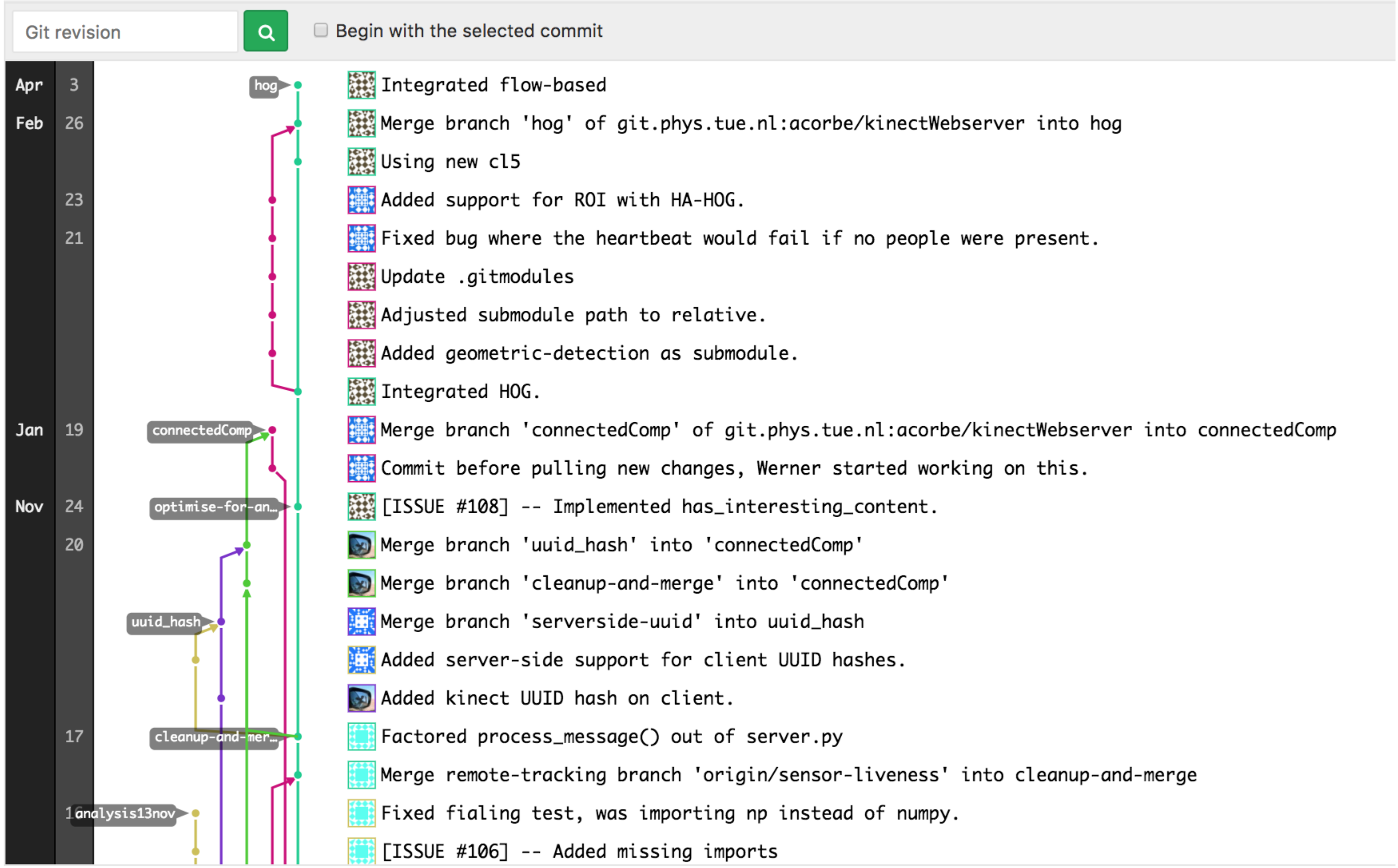
Added support for ROI with HA-HOG.
Werner Kroneman authored 2 months ago

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



Git revision ☐ Begin with the selected commit

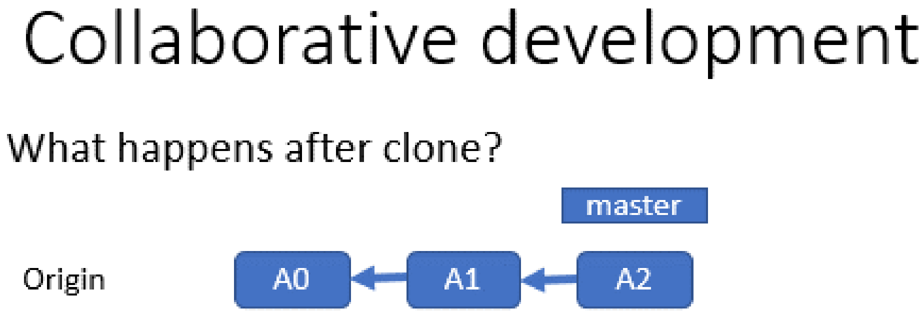
Integrated flow-based
Merge branch 'hog' of git.phys.tue.nl:acorbe/kinectWebserver into hog
Using new cl5
Added support for ROI with HA-HOG.
Fixed bug where the heartbeat would fail if no people were present.
Update .gitmodules
Adjusted submodule path to relative.
Added geometric-detection as submodule.
Integrated HOG.
Merge branch 'connectedComp' of git.phys.tue.nl:acorbe/kinectWebserver into connectedComp
Commit before pulling new changes, Werner started working on this.
[ISSUE #108] -- Implemented has_interesting_content.
Merge branch 'uuid_hash' into 'connectedComp'
Merge branch 'cleanup-and-merge' into 'connectedComp'
Merge branch 'serverside-uuid' into uuid_hash
Added server-side support for client UUID hashes.
Added kinect UUID hash on client.
Factored process_message() out of server.py
Merge remote-tracking branch 'origin/sensor-liveness' into cleanup-and-merge
Fixed fialing test, was importing np instead of numpy.
[ISSUE #106] -- Added missing imports

Interaction between two or more developers

```
git clone / push / pull
```

collaboration building blocks → How to operate them?

Interaction between two or more developers

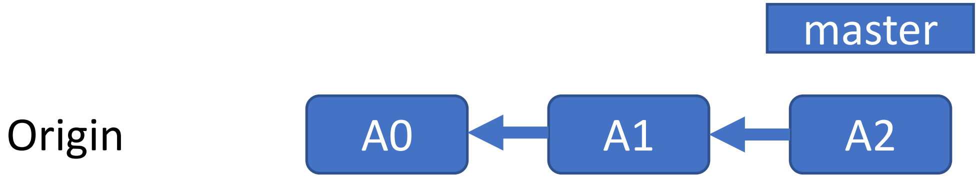


Collaborative development

What happens after clone?

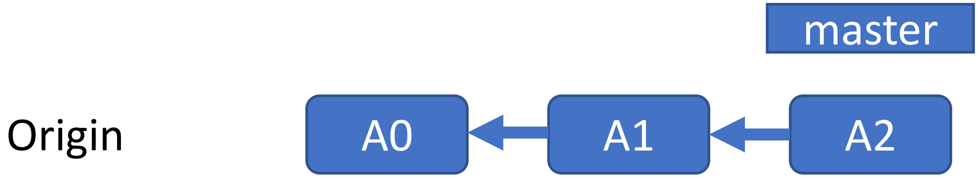
Collaborative development

What happens after clone?



Collaborative development

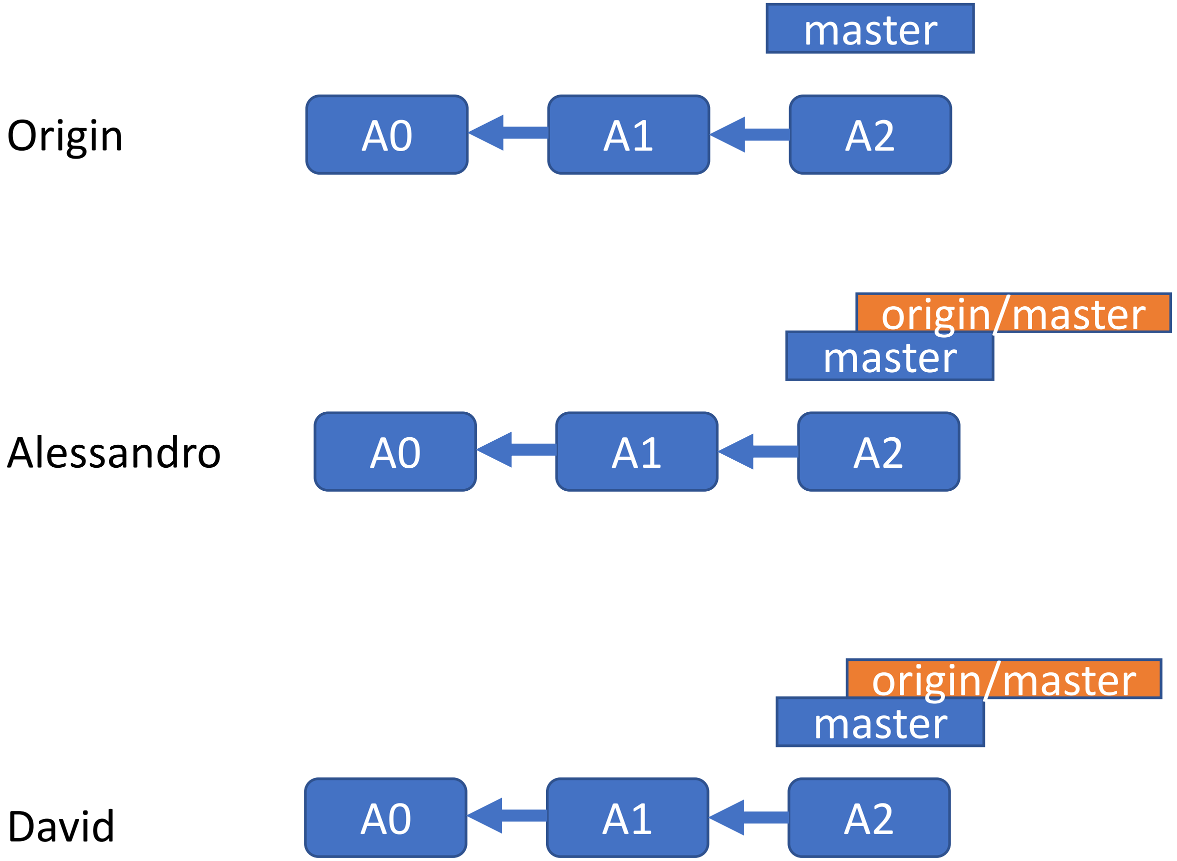
What happens after clone?



Alessandro & David both do `git clone`

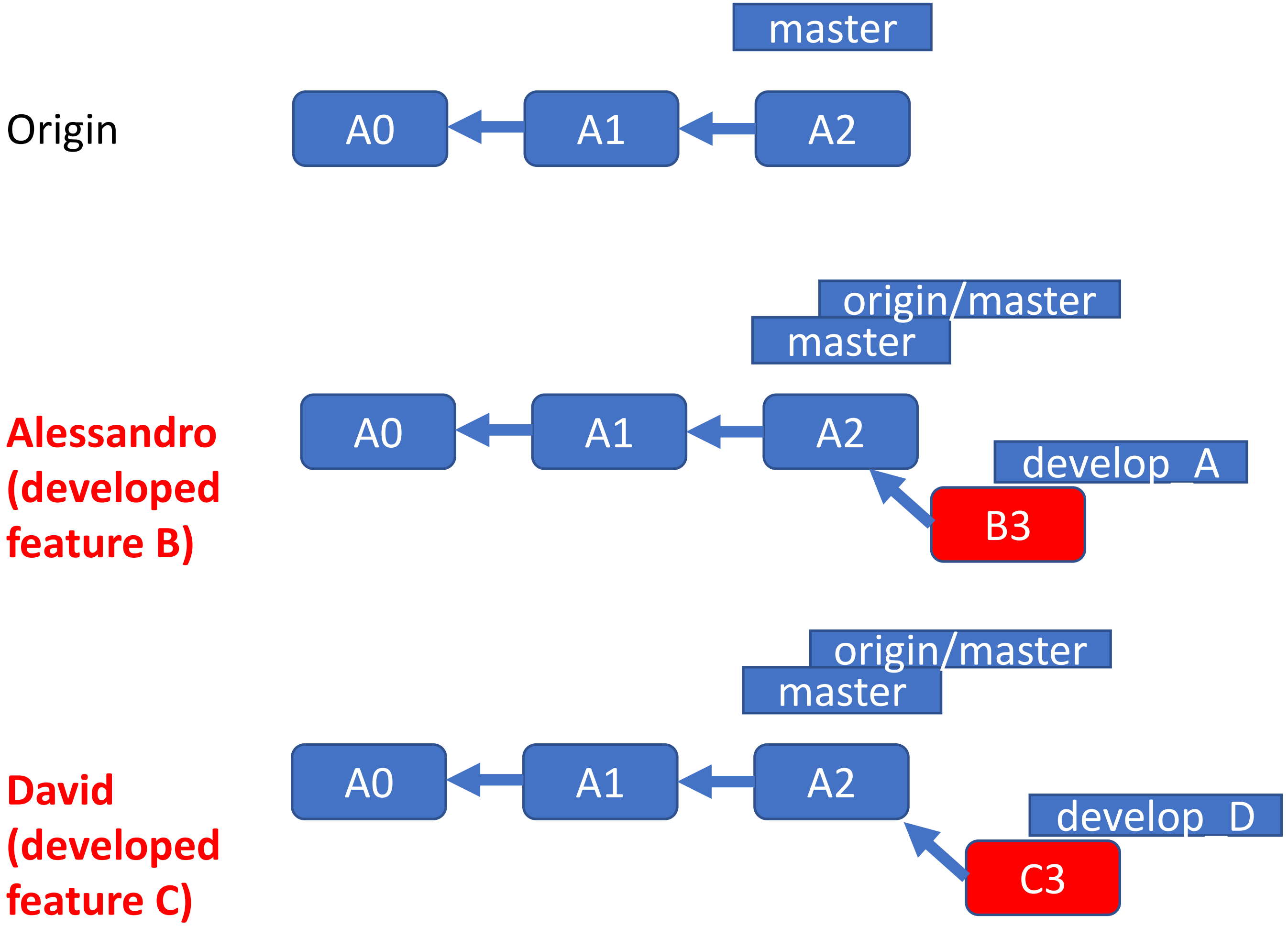
Collaborative development

What happens after clone?



Collaborative development

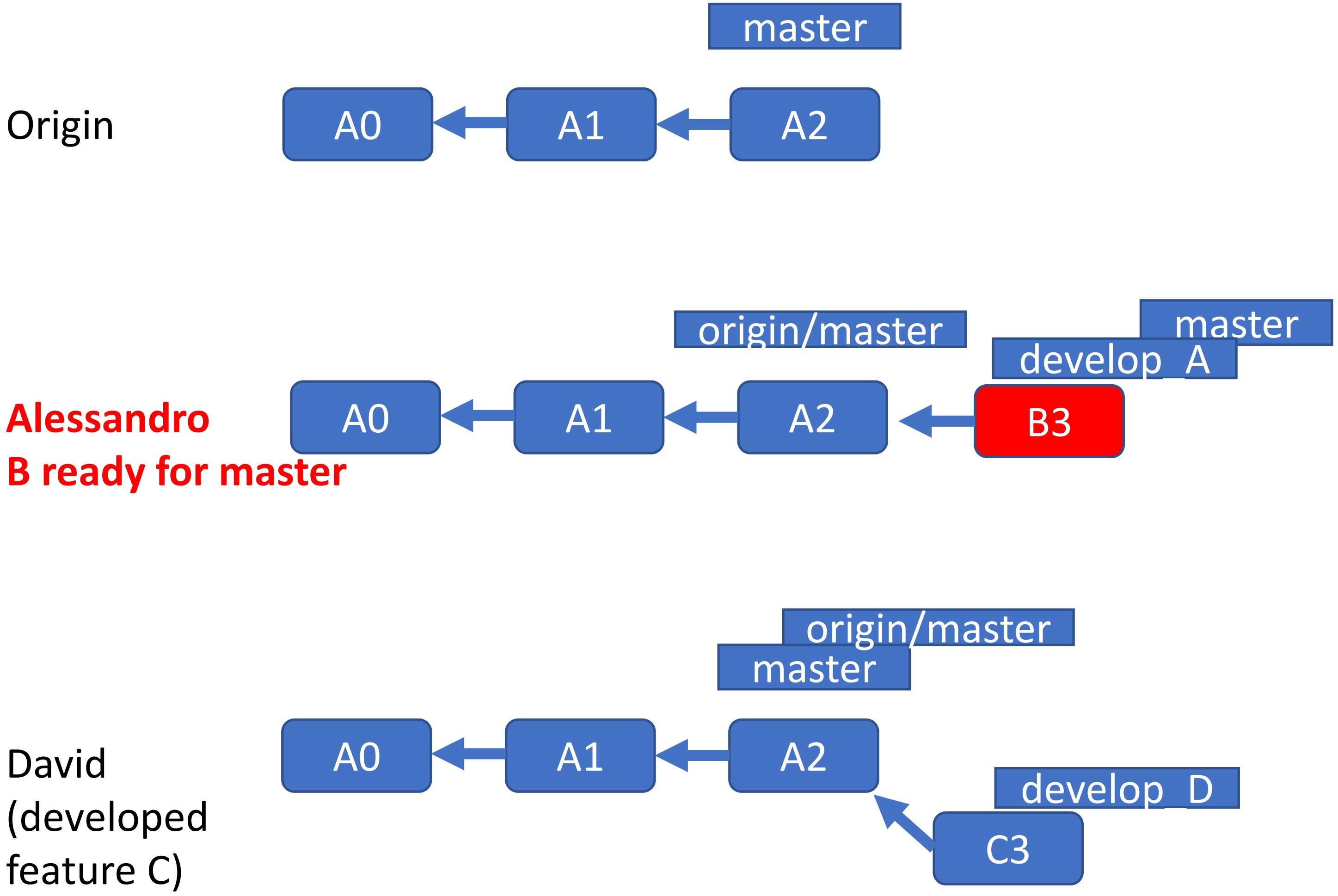
What happens after clone?



Both David and Alessandro committed

Collaborative development

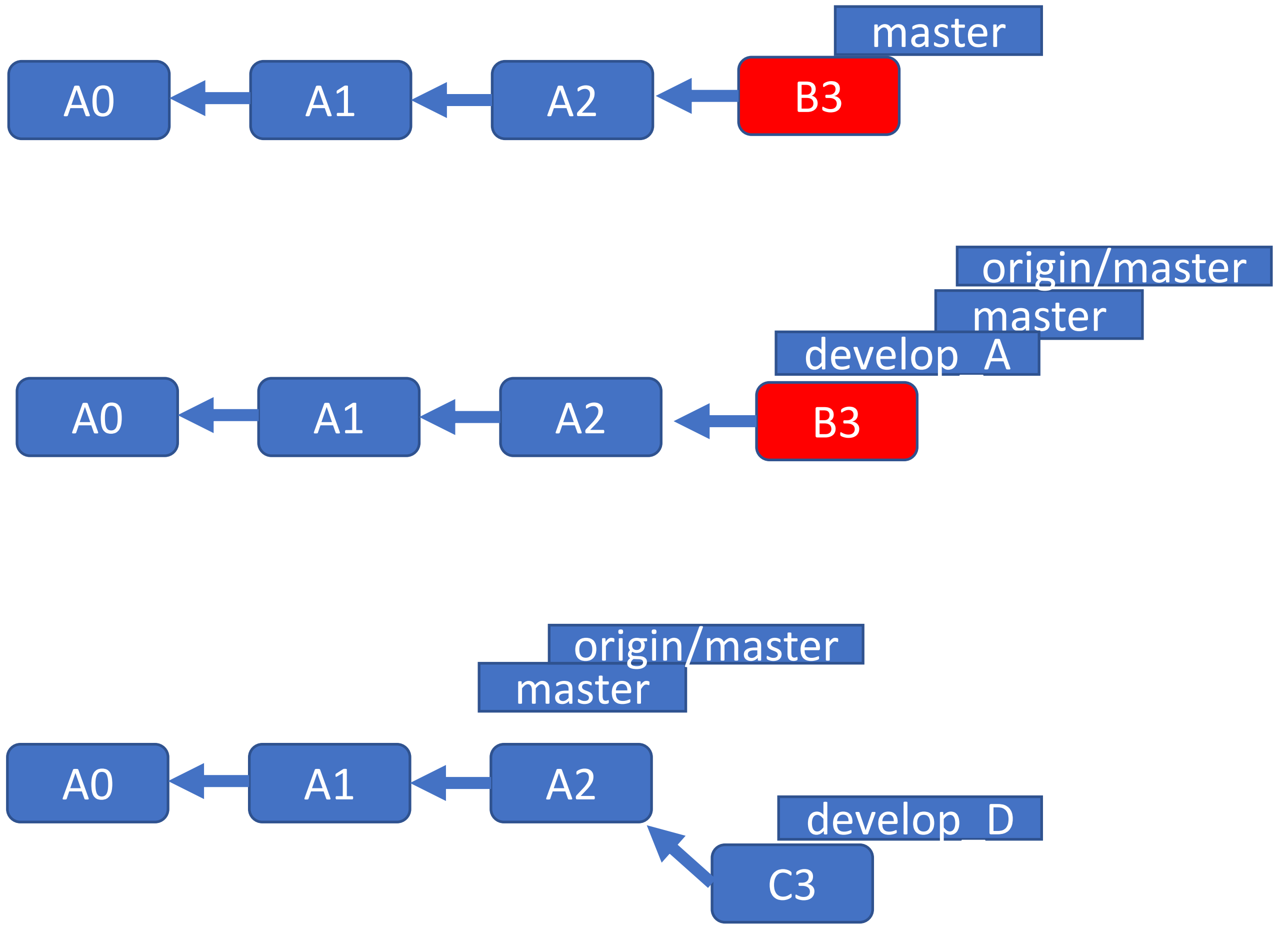
What happens after clone?



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

Collaborative development

What happens after clone?



Origin is updated

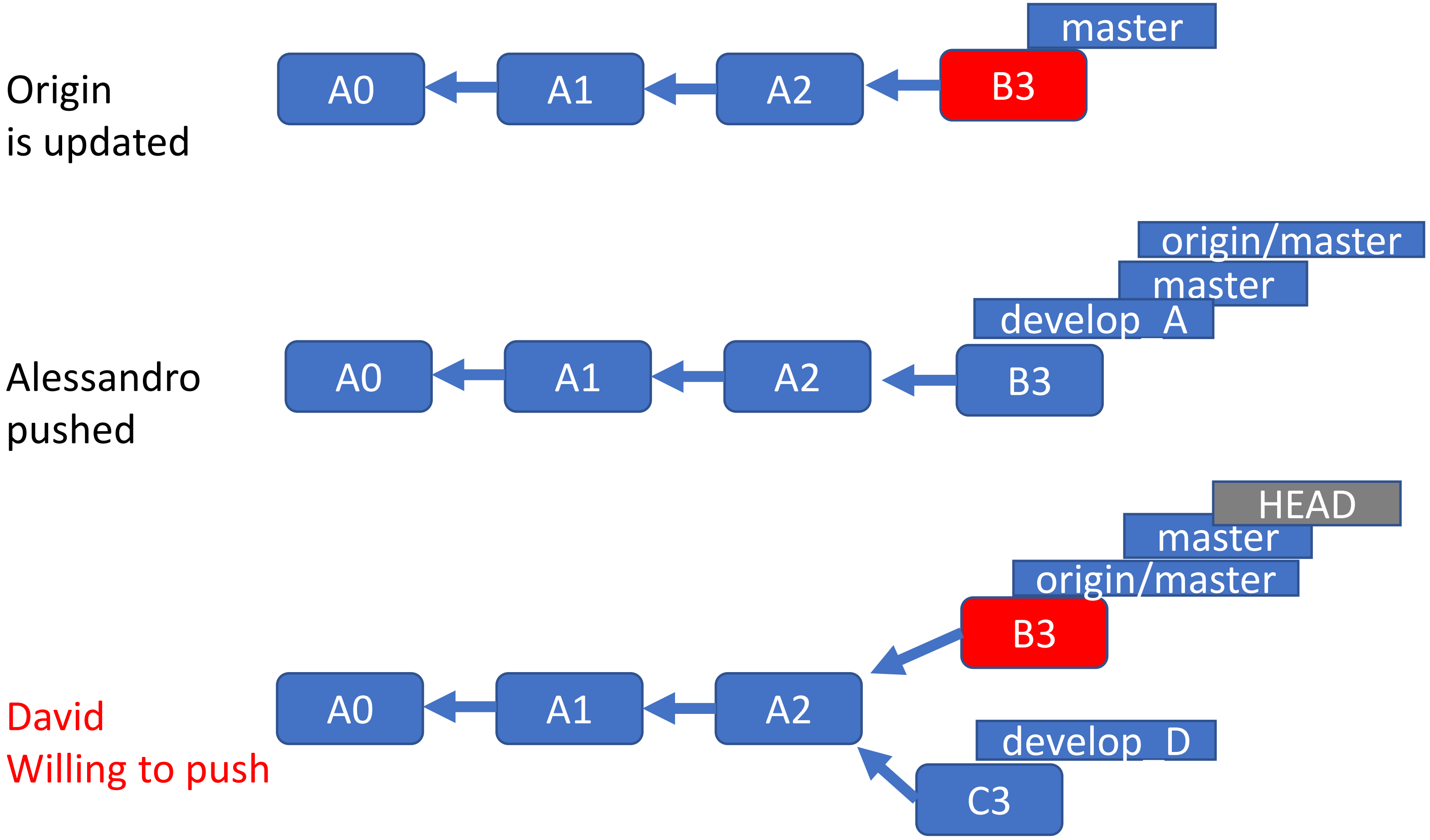
Alessandro pushes B

David (developed feature C)

```
git push origin master
```

Collaborative development

What happens after clone?



Origin
is updated

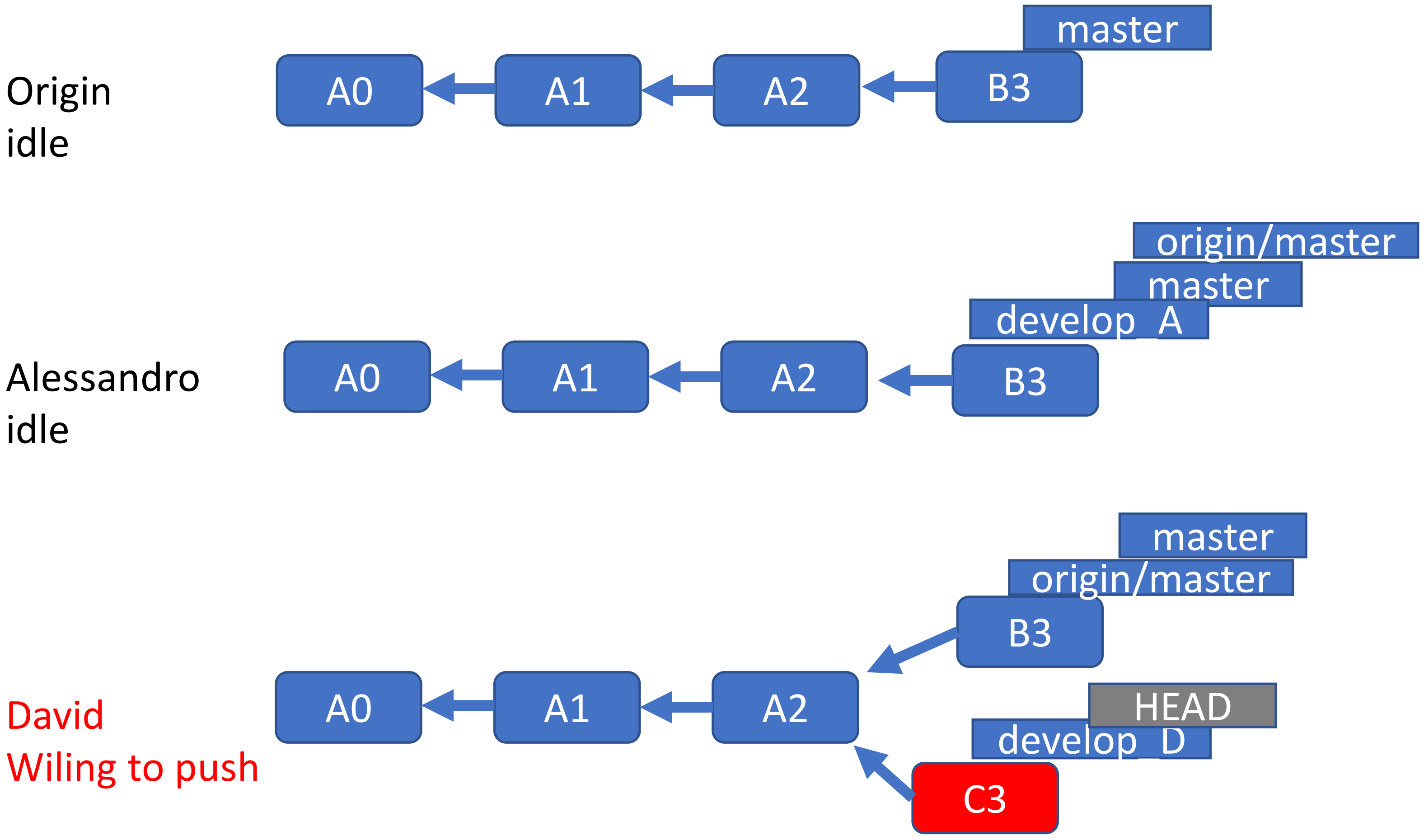
Alessandro
pushed

David
Willing to push

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

Collaborative development

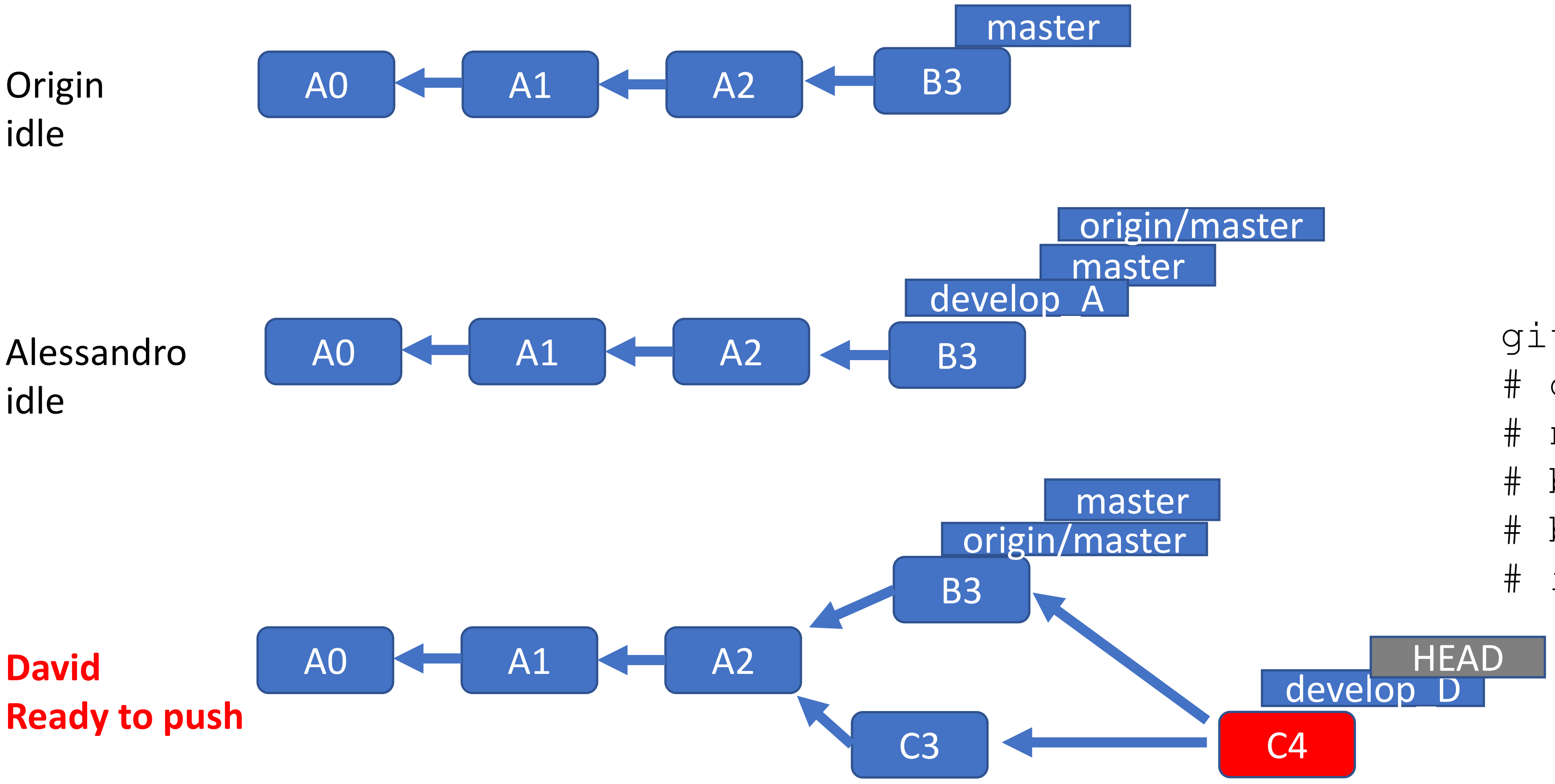
What happens after clone?



```
git checkout develop_D
```

Collaborative development

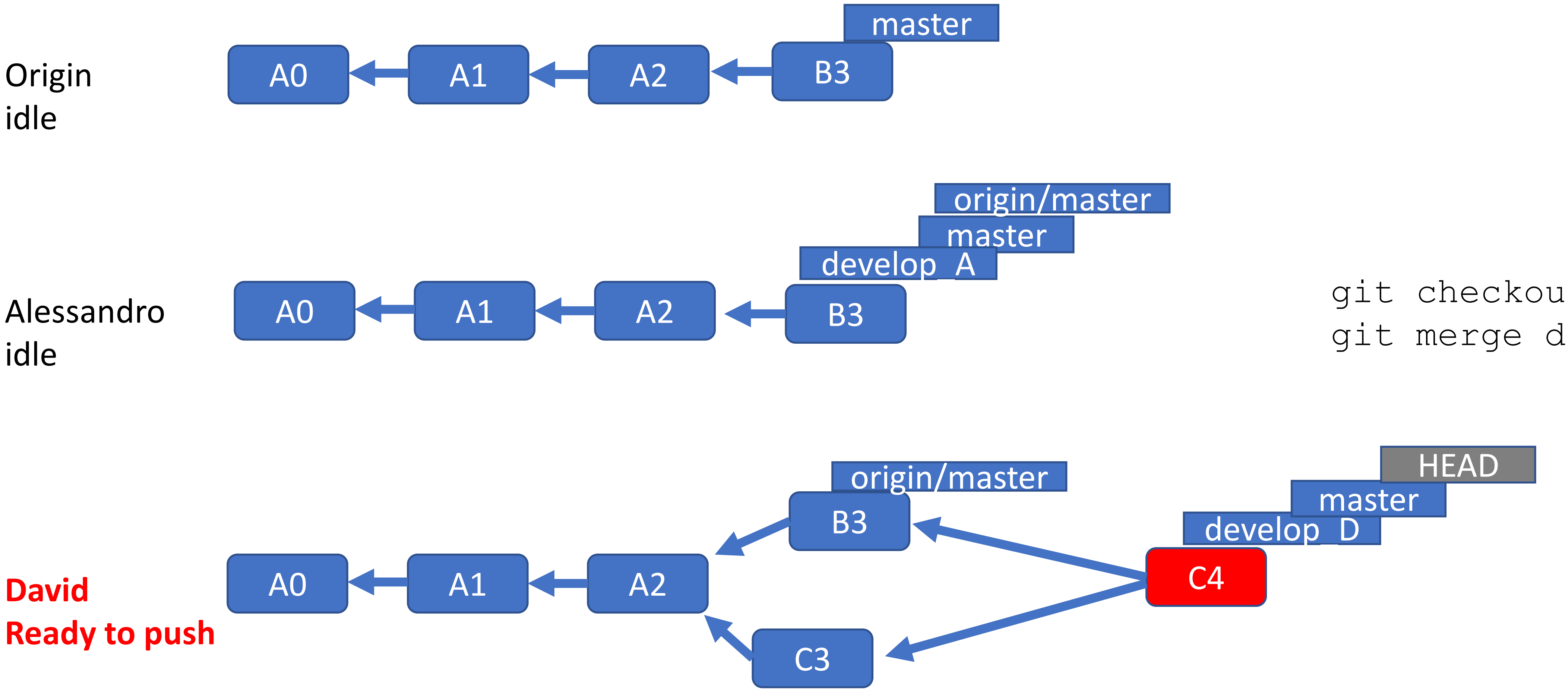
What happens after clone?



```
git merge master
# develop now is
# newer than master
# by definition can
# be safely merged
# into master
```

Collaborative development

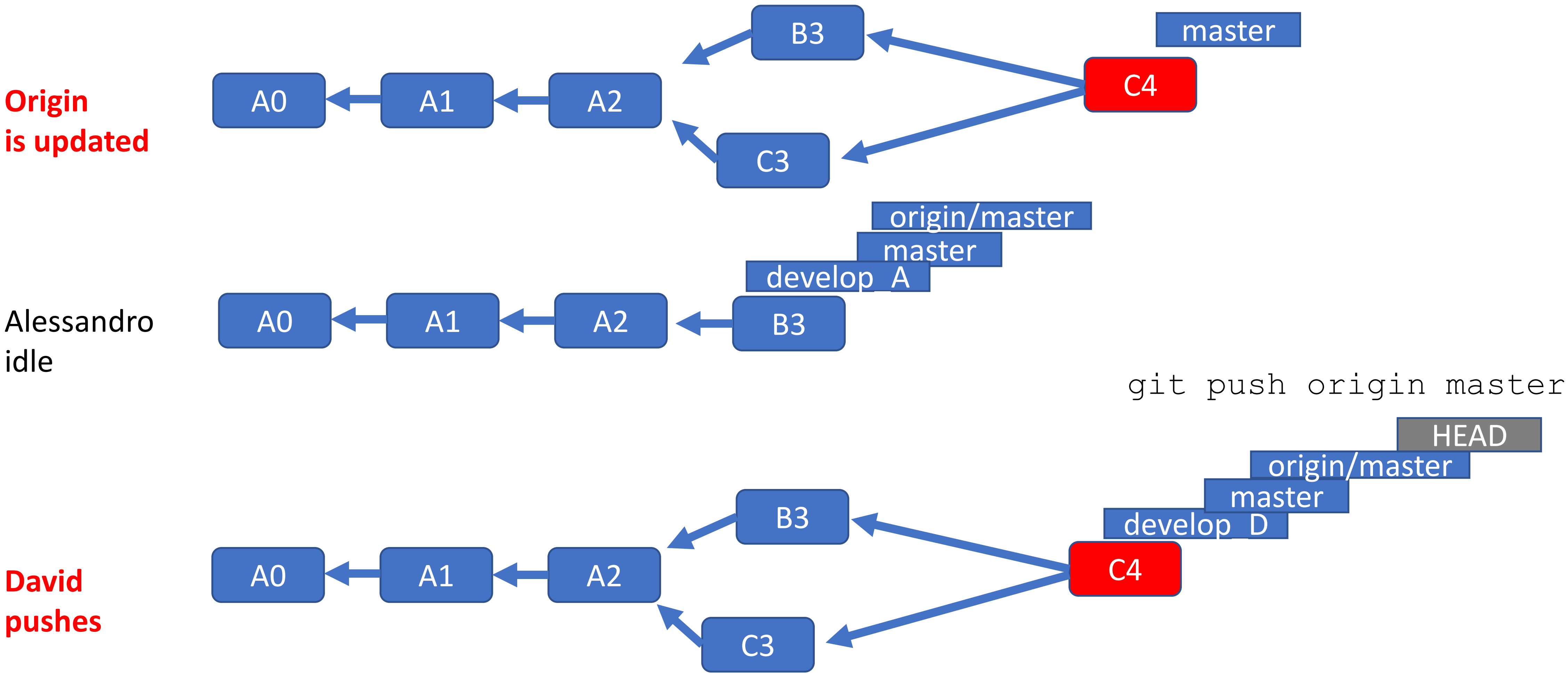
What happens after clone?



```
git checkout master
git merge develop_D
```

Collaborative development

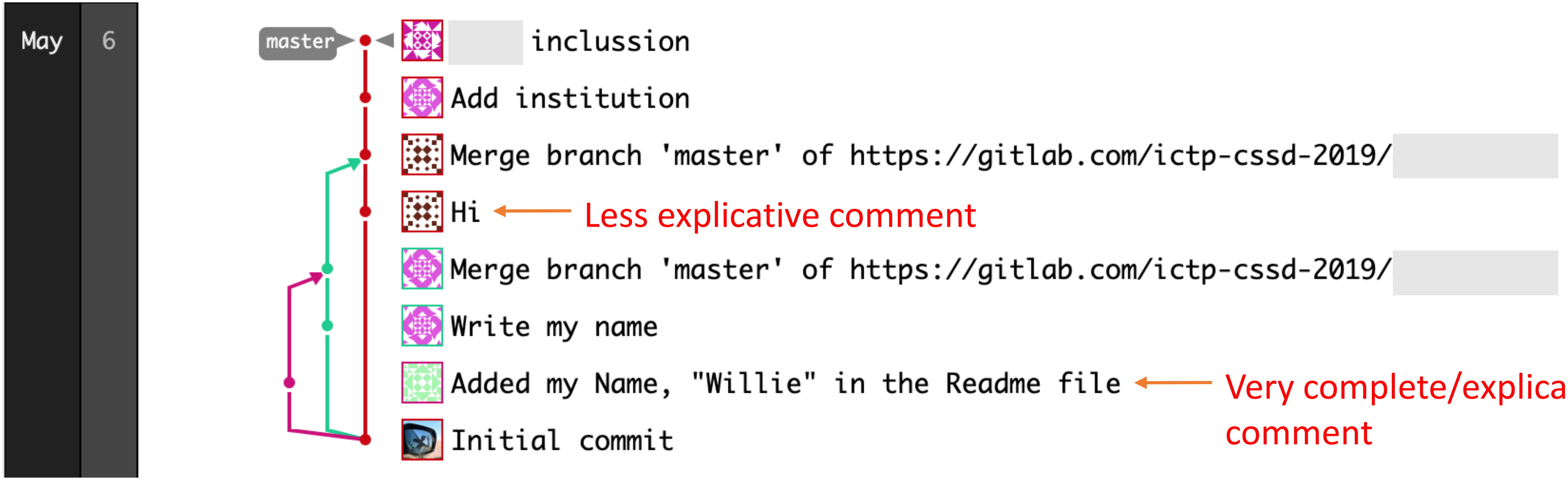
What happens after clone?



```
git push origin master
```

Maximizing scalability and effectiveness

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



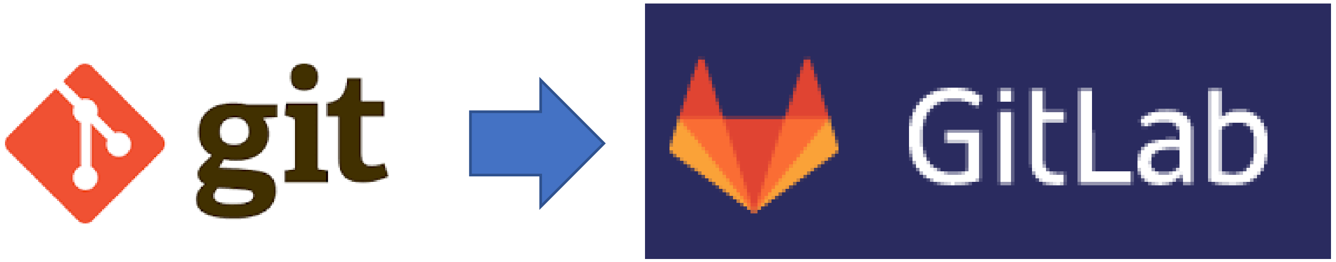
```
inclussion
Add institution
Merge branch 'master' of https://gitlab.com/ictp-cssd-2019/
Hi
Merge branch 'master' of https://gitlab.com/ictp-cssd-2019/
Write my name
Added my Name, "Willie" in the Readme file
Initial commit
```

Less explicative comment

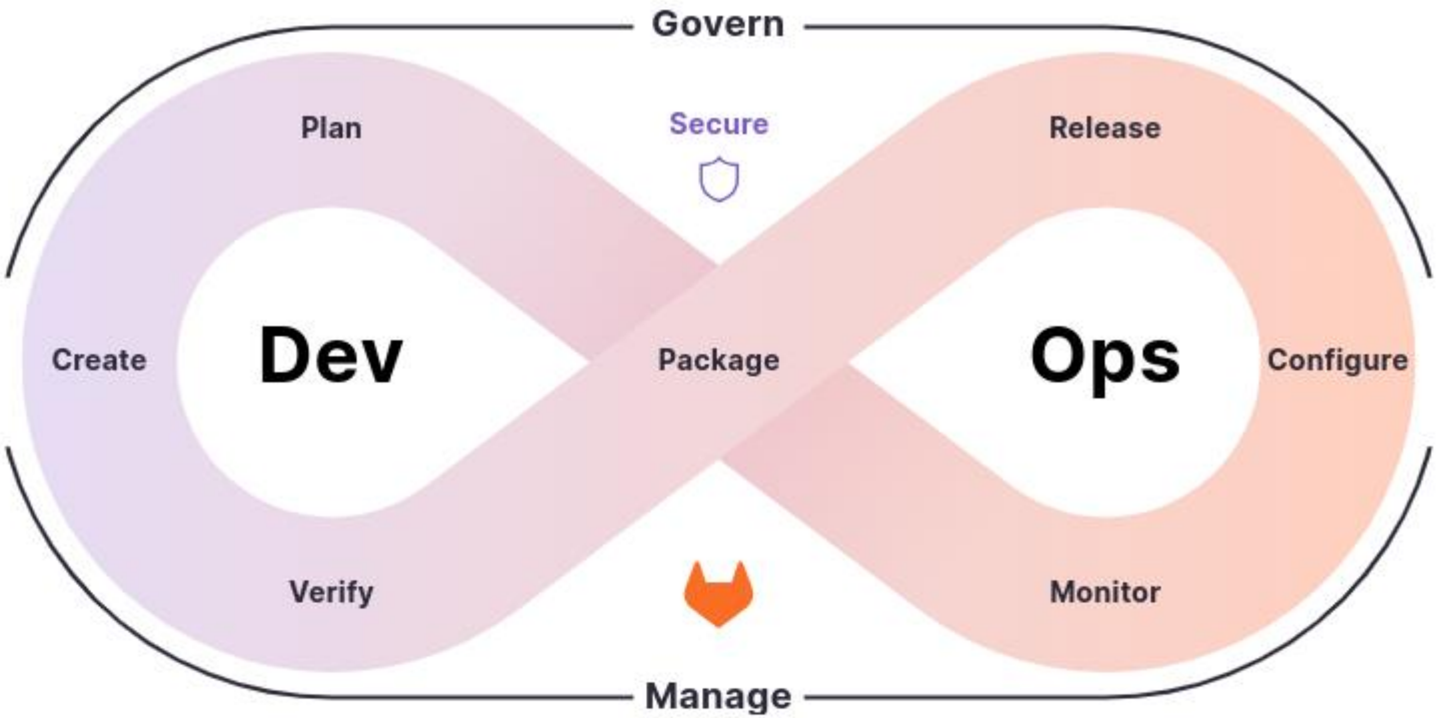
Very complete/explicative comment

Maximizing scalability and effectiveness?

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



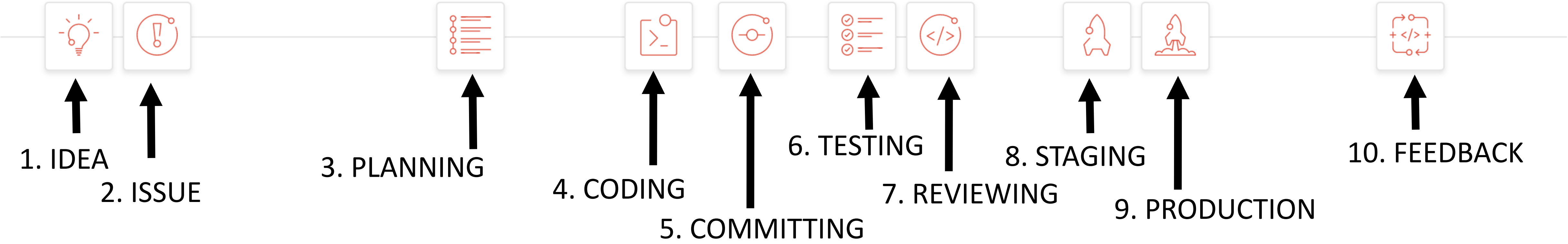
DevOps



DevOps is a combination of software development (dev) & operations (ops).

Software engineering methodology integrating development operations by facilitating collaboration & shared responsibility.

The conversational development paradigm



1. IDEA

2. ISSUE

3. PLANNING

4. CODING

5. COMMITTING

6. TESTING

7. REVIEWING

8. STAGING

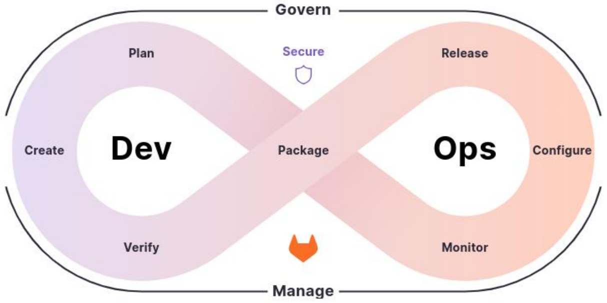
9. PRODUCTION

10. FEEDBACK

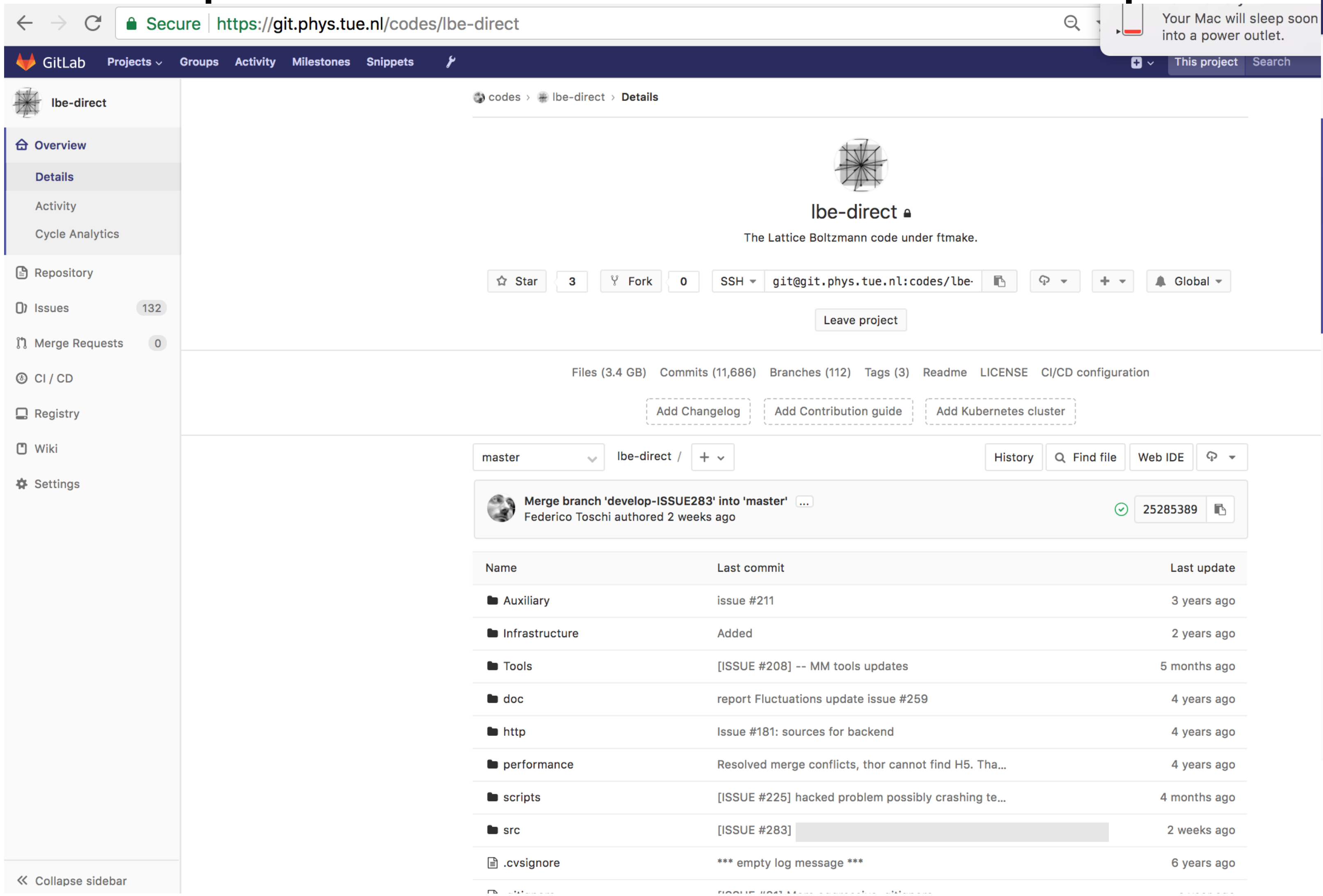
Gitlab/github: meant to support this approach

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



Secure https://git.phys.tue.nl/codes/lbe-direct

Your Mac will sleep soon unless plugged into a power outlet.

GitLab Projects Groups Activity Milestones Snippets — This project Search

lbe-direct

- Overview
  - Details
  - Activity
  - Cycle Analytics
- Repository
- Issues 132
- Merge Requests 0
- CI / CD
- Registry
- Wiki
- Settings

Collapse sidebar

codes › lbe-direct › Details

lbe-direct

The Lattice Boltzmann code under ftmake.

Star 3 Fork 0 SSH git@git.phys.tue.nl:codes/lbe- Global

Leave project

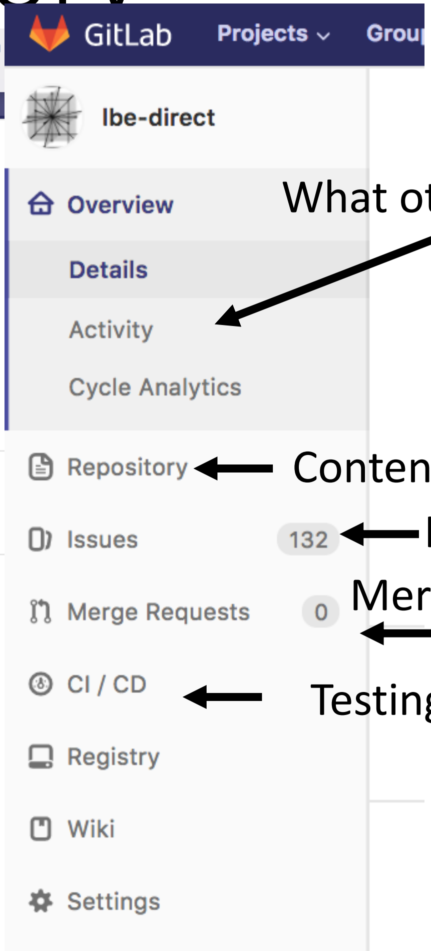
Files (3.4 GB) Commits (11,686) Branches (112) Tags (3) Readme LICENSE CI/CD configuration

Add Changelog Add Contribution guide Add Kubernetes cluster

master lbe-direct / History Find file Web IDE

Merge branch 'develop-ISSUE283' into 'master' ...
Federico Toschi authored 2 weeks ago 25285389

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] [redacted]	2 weeks ago
.cvsignore	*** empty log message ***	6 years ago



GitLab Projects Grou

lbe-direct

- Overview
  - Details
  - Activity
  - Cycle Analytics
- Repository
- Issues 132
- Merge Requests 0
- CI / CD
- Registry
- Wiki
- Settings

What others developer did?

Content/commits

Issues

Merging your contribution

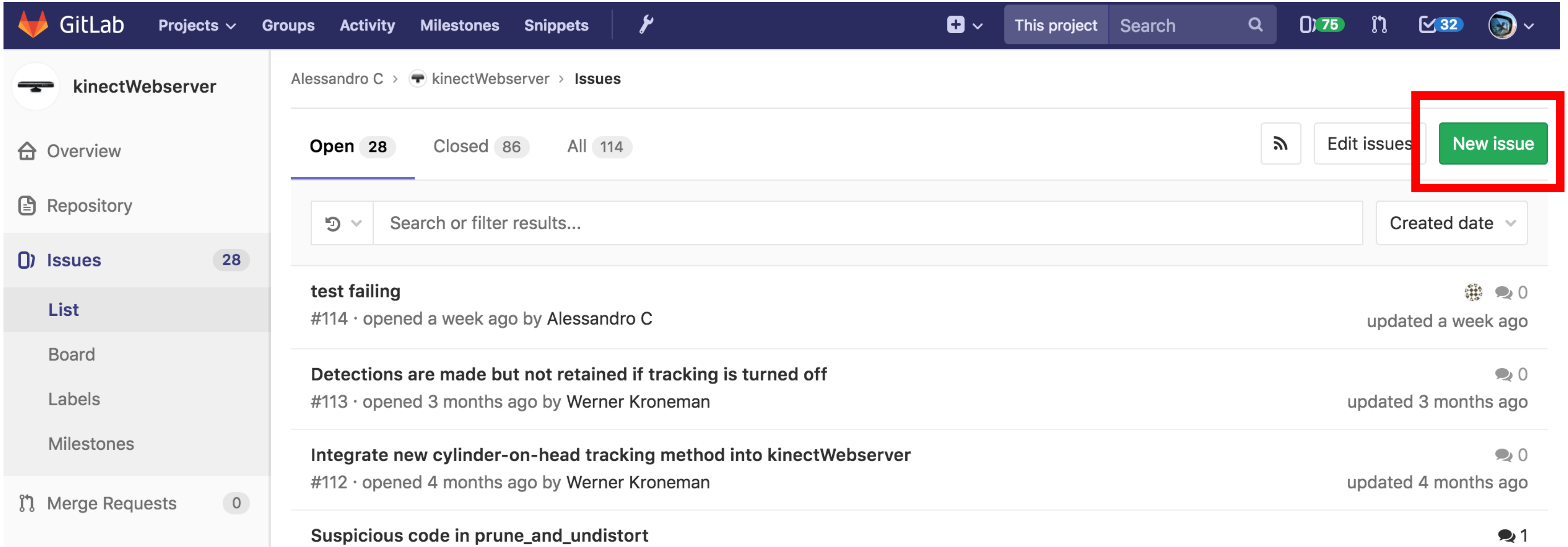
Testing

Issue first

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

Issue first

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



test failing 0
#114 · opened a week ago by Alessandro C — updated a week ago

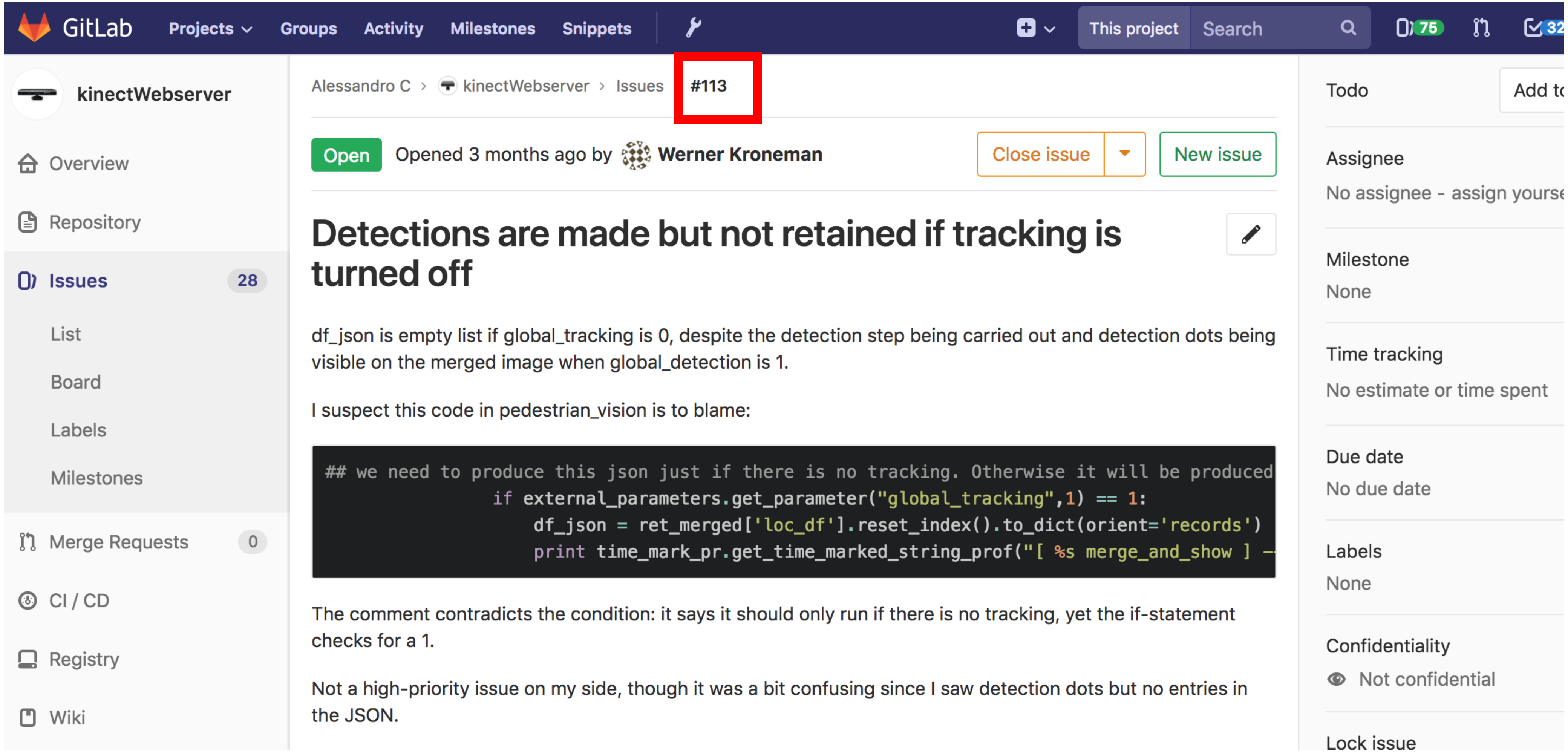
Detections are made but not retained if tracking is turned off 0
#113 · opened 3 months ago by Werner Kroneman — updated 3 months ago

Integrate new cylinder-on-head tracking method into kinectWebserver 0
#112 · opened 4 months ago by Werner Kroneman — updated 4 months ago

Suspicious code in prune_and_undistort 1

Issue at first

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



Alessandro C › kinectWebserver › Issues #113

Open Opened 3 months ago by Werner Kroneman

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.

Assignee
No assignee - assign yourse

Milestone
None

Time tracking
No estimate or time spent

Due date
No due date

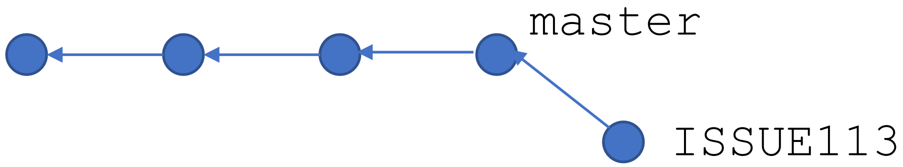
Labels
None

Confidentiality
Not confidential

Lock issue

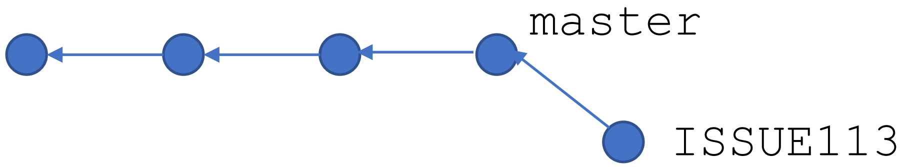
From issue to code

- 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

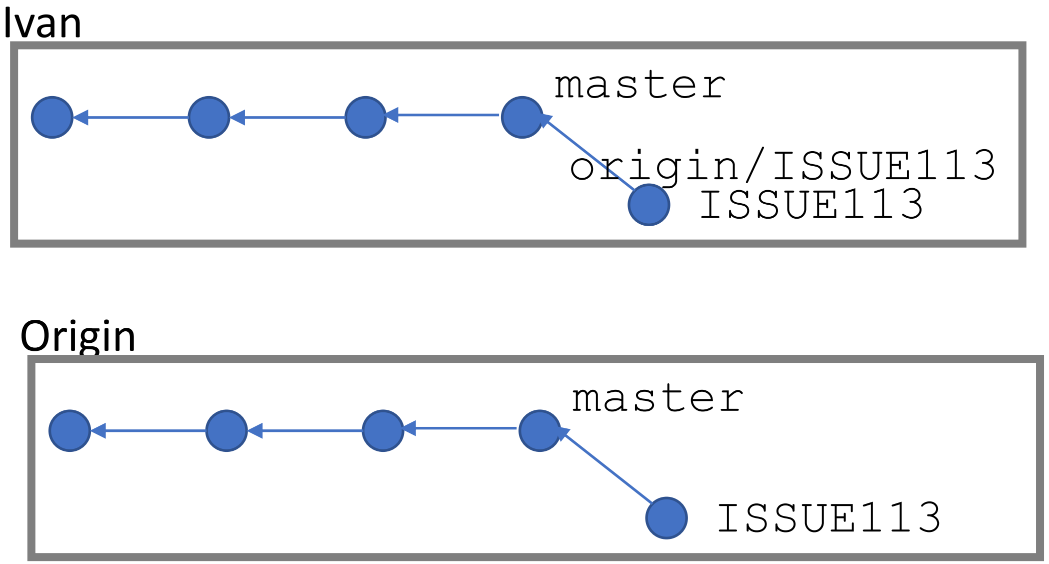
- If Ivan is in charge of fixing #113:
  - [Clones the code – if he does not have already]
  - Branches master -> e.g. into branch ISSUE113 (for trackability)
  - `git checkout –b ISSUE113`



  - 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

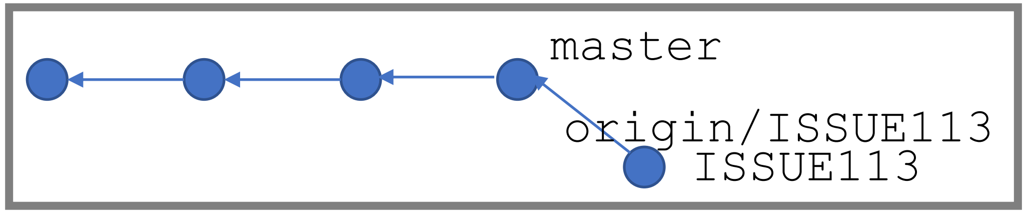
- `git commit –m ‘[ISSUE #113] – how code and tests have been changed’`
- `git push origin ISSUE113`



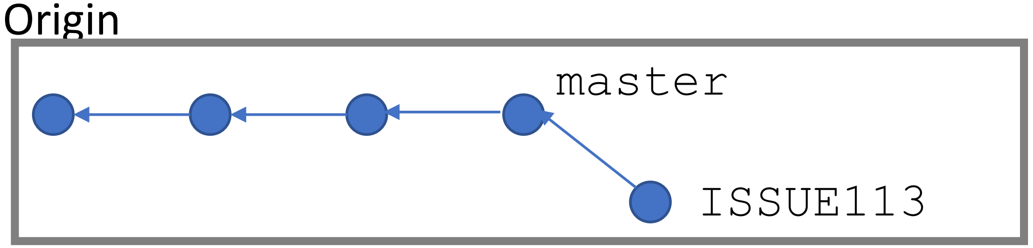
From issue to code

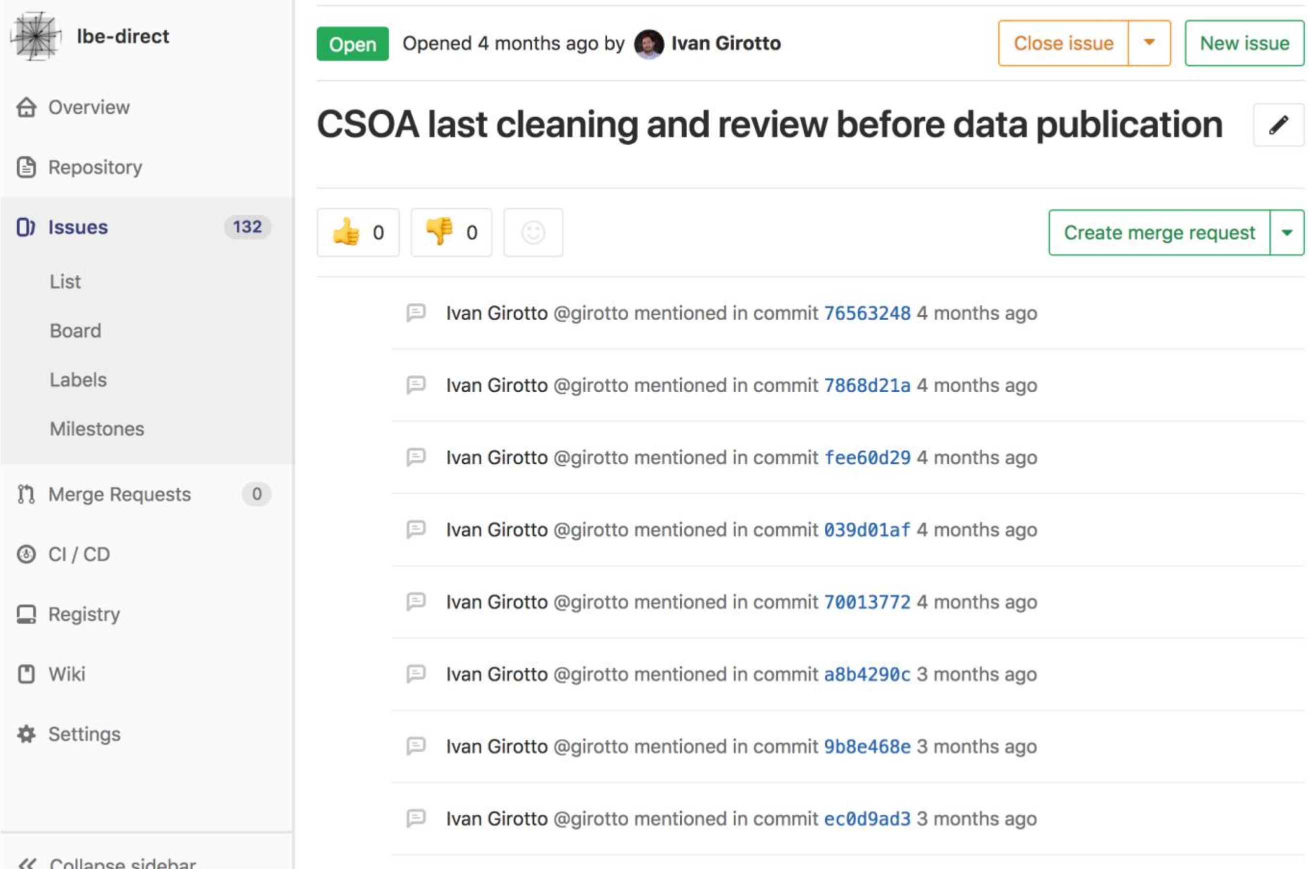
- git commit -m ‘[ISSUE #113] – how code and tests have been changed’
- git push origin ISSUE113

Ivan



Origin





Open Opened 4 months ago by Ivan Girotto

CSOA last cleaning and review before data publication

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

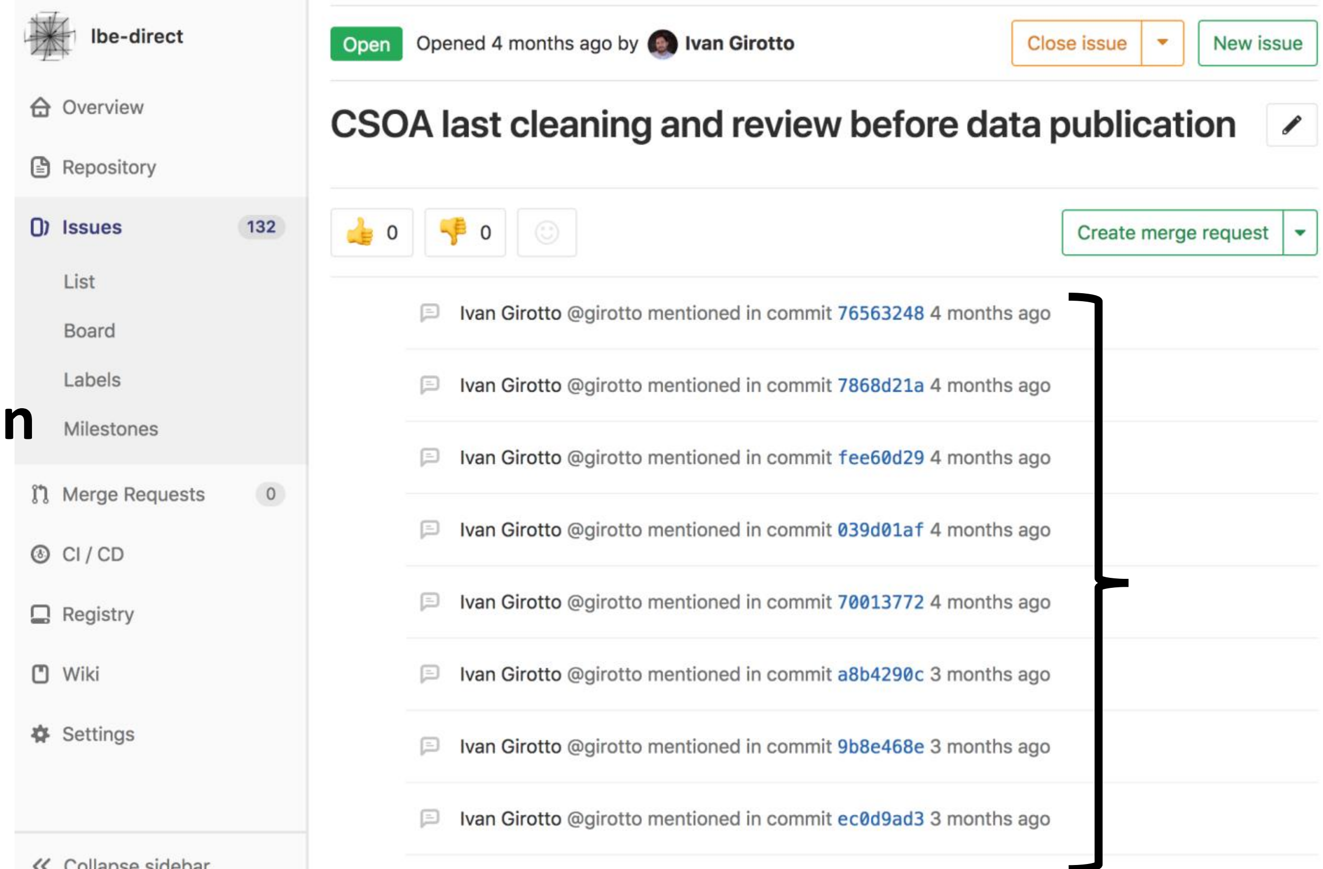
Ivan Girotto @girotto mentioned in commit 9b8e468e 3 months ago

Ivan Girotto @girotto mentioned in commit ec0d9ad3 3 months ago

From issue to code

```
• 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**



Open Opened 4 months ago by Ivan Girotto

CSOA last cleaning and review before data publication

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

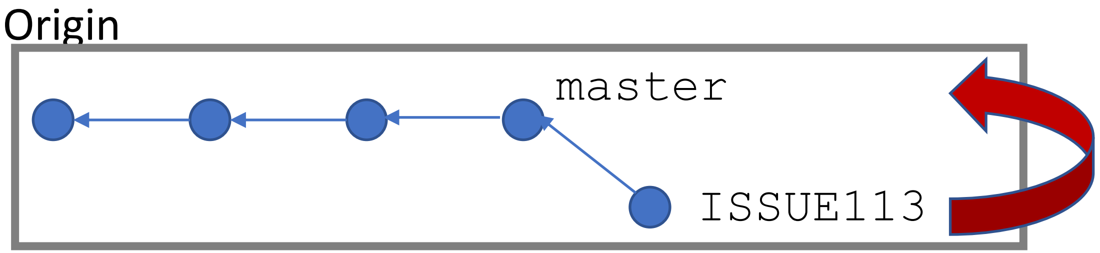
Ivan Girotto @girotto mentioned in commit 70013772 4 months ago

Ivan Girotto @girotto mentioned in commit a8b4290c 3 months ago

Ivan Girotto @girotto mentioned in commit 9b8e468e 3 months ago

Ivan Girotto @girotto mentioned in commit ec0d9ad3 3 months ago

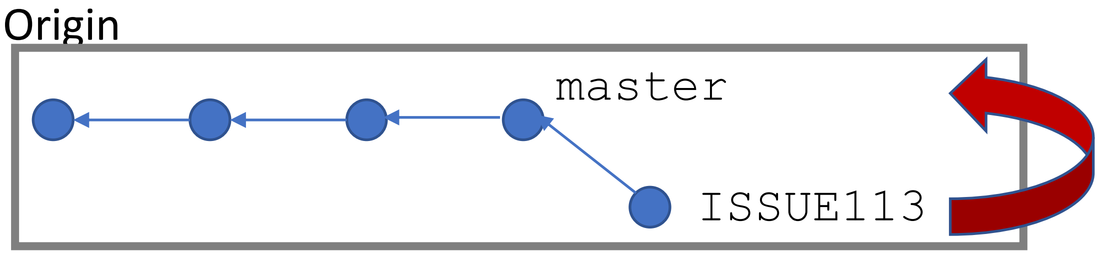
How to contribute to master?



```
git push origin ISSUE113
```

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

How to contribute to master?



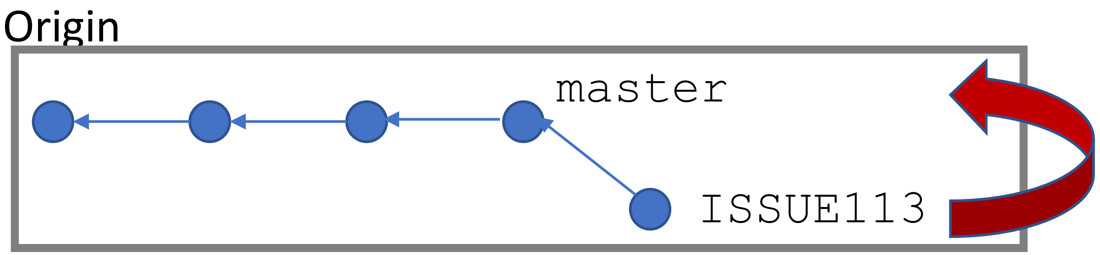
```
git push origin ISSUE113
```

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

- **Required: code review from authoritative parties**
- Ask authority to **merge** contribution **through <u>merge request</u>**

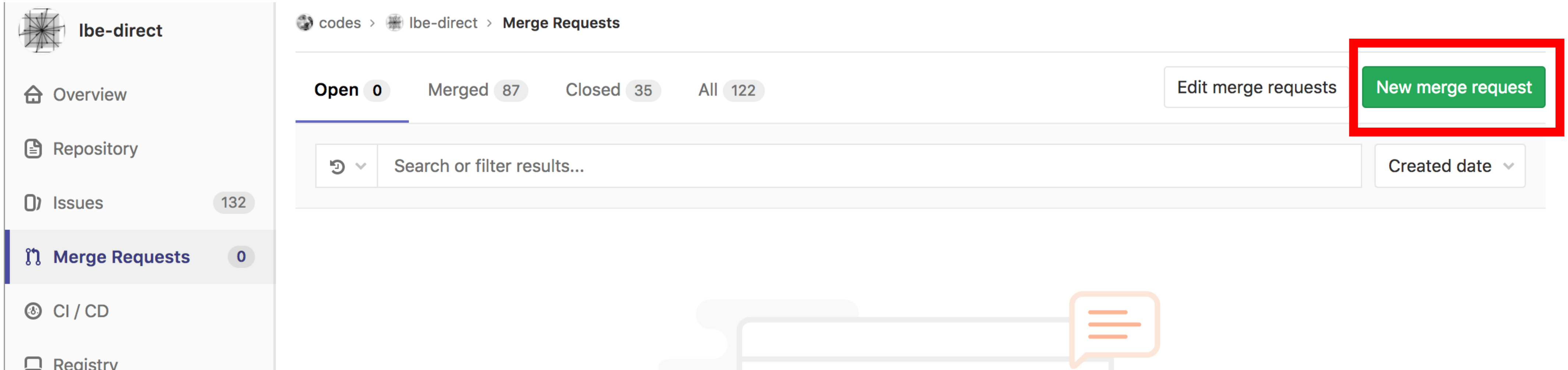
Code review merge request

```
git push origin ISSUE113
```



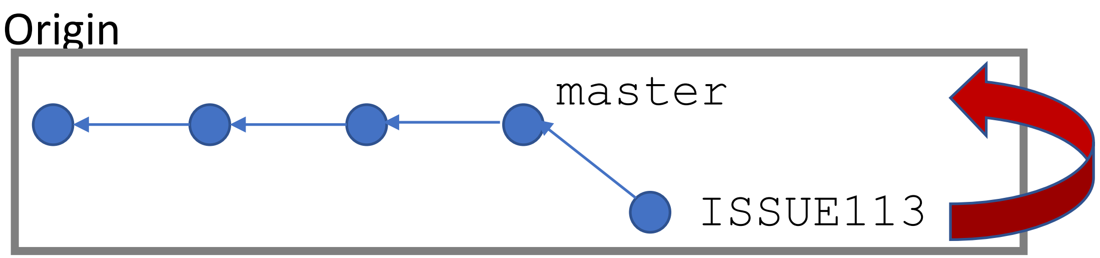
master is a privileged branch.
Developers must not push directly! (generally prohibited)

- **Required: code review from authoritative parties**
- Ask authority to **merge** contribution **through merge request**



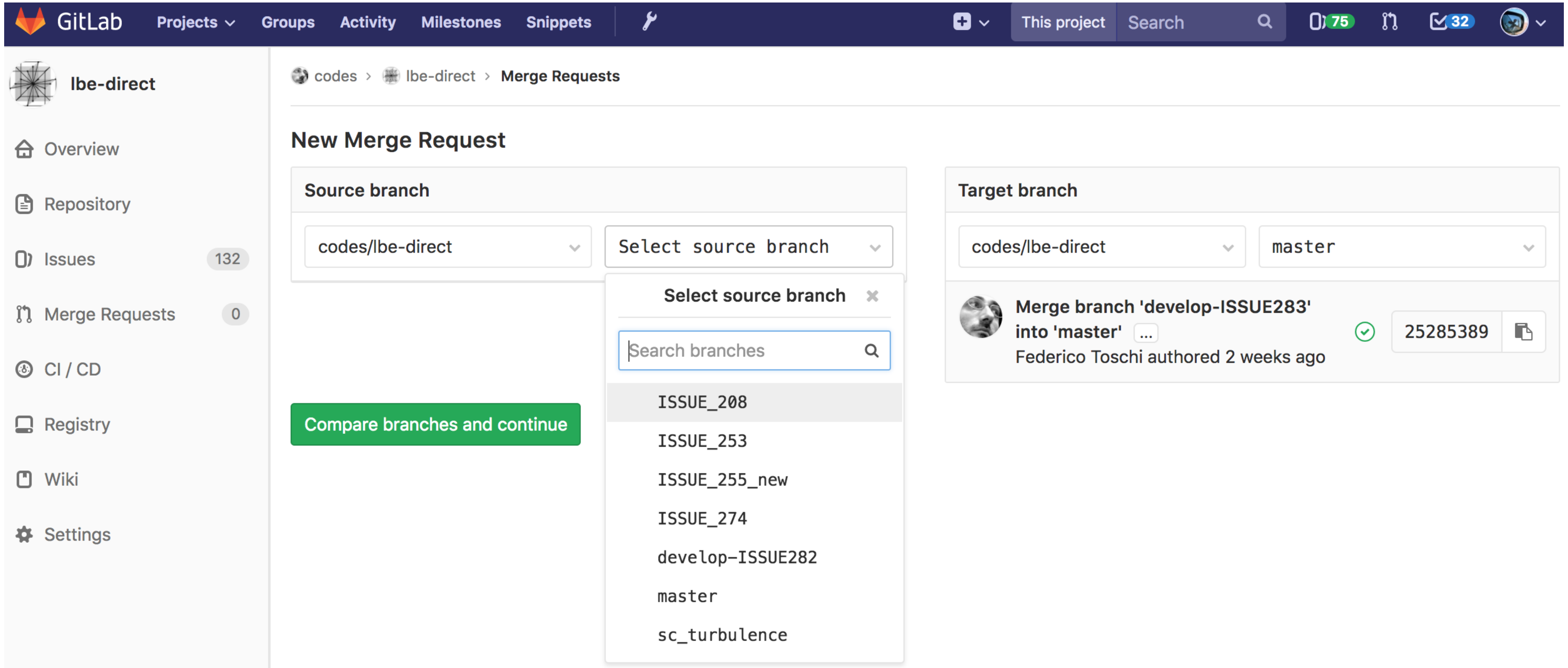
Code review merge request

```
git push origin ISSUE113
```



master is a privileged branch.

Developers must not push directly! (generally prohibited)



New Merge Request

Source branch

codes/lbe-direct

Select source branch

Select source branch

Search branches

ISSUE_208
ISSUE_253
ISSUE_255_new
ISSUE_274
develop-ISSUE282
master
sc_turbulence

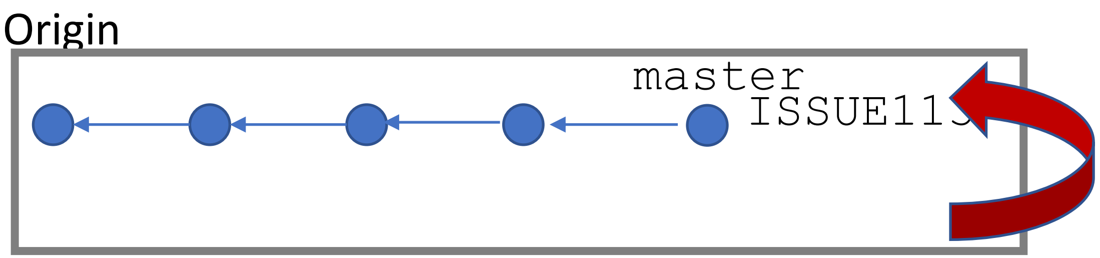
Target branch

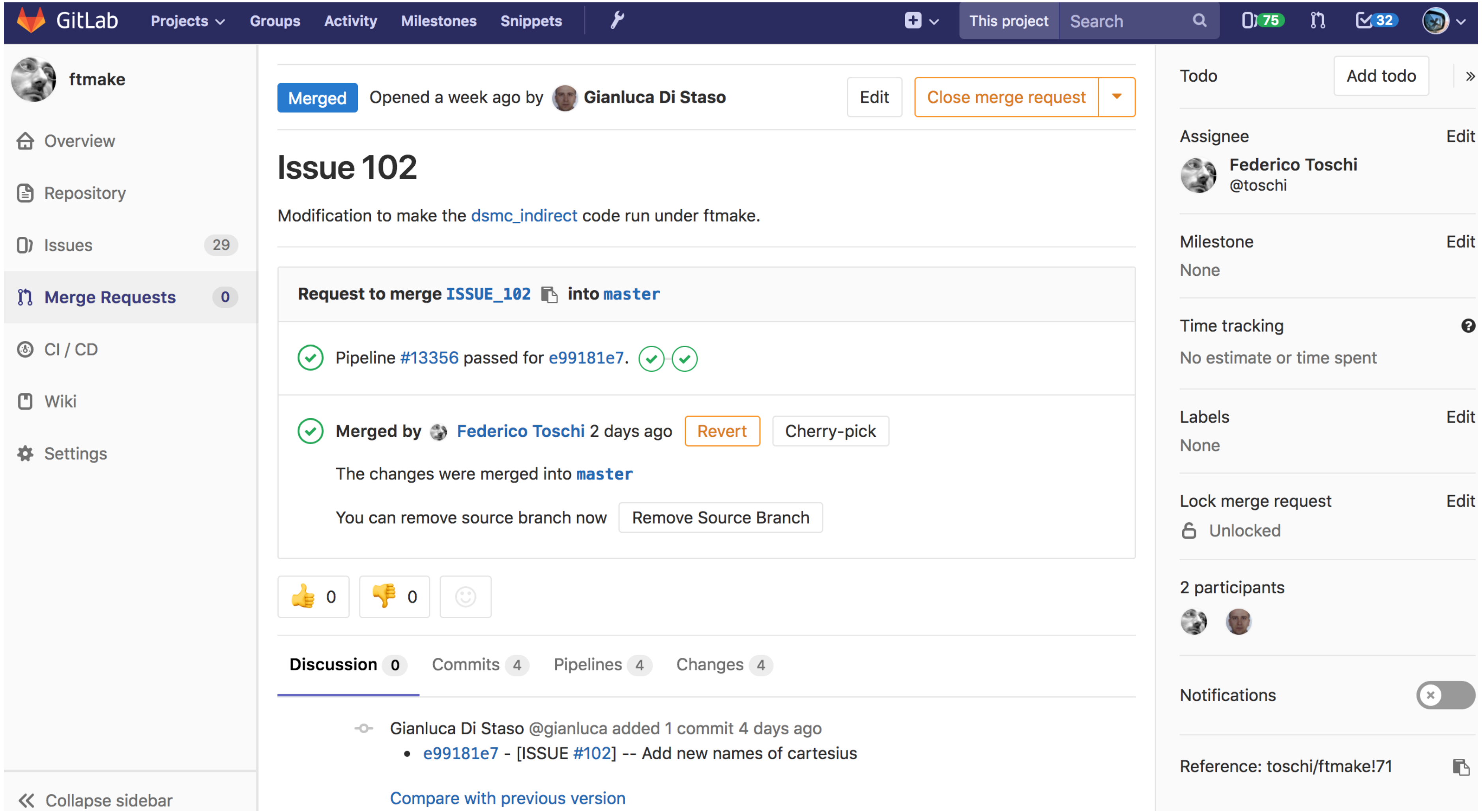
codes/lbe-direct

master

Merge branch 'develop-ISSUE283' into 'master' ...
Federico Toschi authored 2 weeks ago
25285389

Code review merge request





ftmake

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

Collapse sidebar

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 cartesius

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

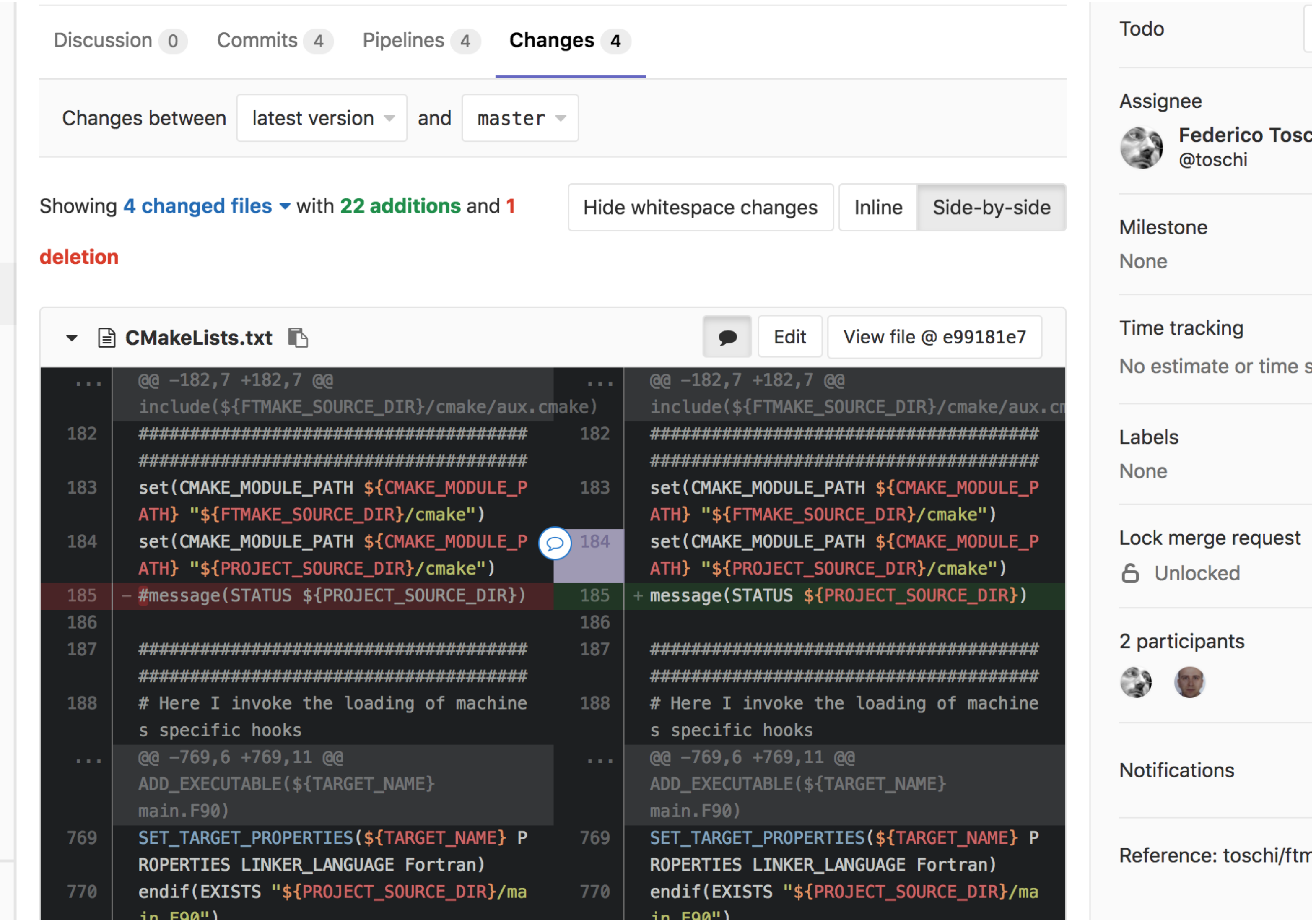
2 participants

Notifications

Reference: toschi/ftmake!71

Code review merge request

Code must be reviewed before merging by an authority



Changes between latest version and master

Showing 4 changed files with 22 additions and 1 deletion

CMakeLists.txt

```
@@ -182,7 +182,7 @@
include(${FTMAKE_SOURCE_DIR}/cmake/aux.cmake)
######################################
######################################
set(CMAKE_MODULE_PATH ${CMAKE_MODULE_P
ATH} "${FTMAKE_SOURCE_DIR}/cmake")
set(CMAKE_MODULE_PATH ${CMAKE_MODULE_P
ATH} "${PROJECT_SOURCE_DIR}/cmake")
- #message(STATUS ${PROJECT_SOURCE_DIR})

######################################
######################################
# Here I invoke the loading of machine
s specific hooks
@@ -769,6 +769,11 @@
ADD_EXECUTABLE(${TARGET_NAME}
main.F90)
SET_TARGET_PROPERTIES(${TARGET_NAME} P
ROPERTIES LINKER_LANGUAGE Fortran)
endif(EXISTS "${PROJECT_SOURCE_DIR}/ma
in.F90")
```

```
@@ -182,7 +182,7 @@
include(${FTMAKE_SOURCE_DIR}/cmake/aux.cm
######################################
######################################
set(CMAKE_MODULE_PATH ${CMAKE_MODULE_P
ATH} "${FTMAKE_SOURCE_DIR}/cmake")
set(CMAKE_MODULE_PATH ${CMAKE_MODULE_P
ATH} "${PROJECT_SOURCE_DIR}/cmake")
+ message(STATUS ${PROJECT_SOURCE_DIR})

######################################
######################################
# Here I invoke the loading of machine
s specific hooks
@@ -769,6 +769,11 @@
ADD_EXECUTABLE(${TARGET_NAME}
main.F90)
SET_TARGET_PROPERTIES(${TARGET_NAME} P
ROPERTIES LINKER_LANGUAGE Fortran)
endif(EXISTS "${PROJECT_SOURCE_DIR}/ma
in.F90")
```

Todo

Assignee
Federico Tosc
@toschi

Milestone
None

Time tracking
No estimate or time s

Labels
None

Lock merge request
Unlocked

2 participants

Notifications

Reference: toschi/ftm

Structured collaboration

- Git for version control
- Developers contribute by (merging) separate branches
- Developers collaborate via **ISSUES**
- Interaction with «code authority» via merge **requests**
- Commit messages are expressive and related to ISSUES

- Demonstration: after lunch

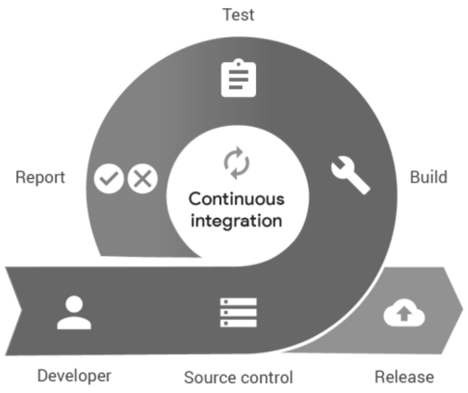
Plan of the lectures

Structured collaboration

Gitlab
Conversational development
DevOps

Trusting changes

Testing (Ali)
Continuous integration
Continuous deployment



Immediate reproducibility

Containers
Docker
Virtualenvs

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(10min))

Also

- Deploy frequently – “agile” dev-user cycle

Coding: collaborative, **trustworthy**, reproducible

Alessandro Corbetta

Plan of the lectures

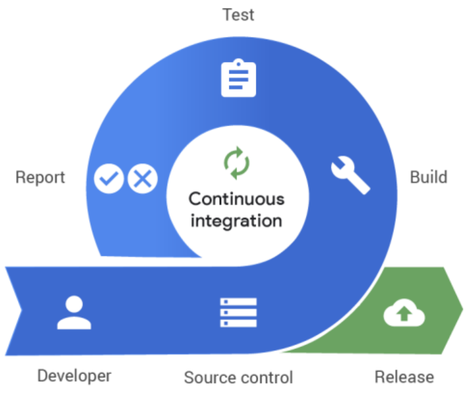
Structured collaboration

Trusting changes

Immediate reproducibility

Gitlab
Conversational development
DevOps

Testing
Continuous integration
Continuous deployment



Containers
Docker
Virtualenvs

Maximizing scalability and effectiveness?

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

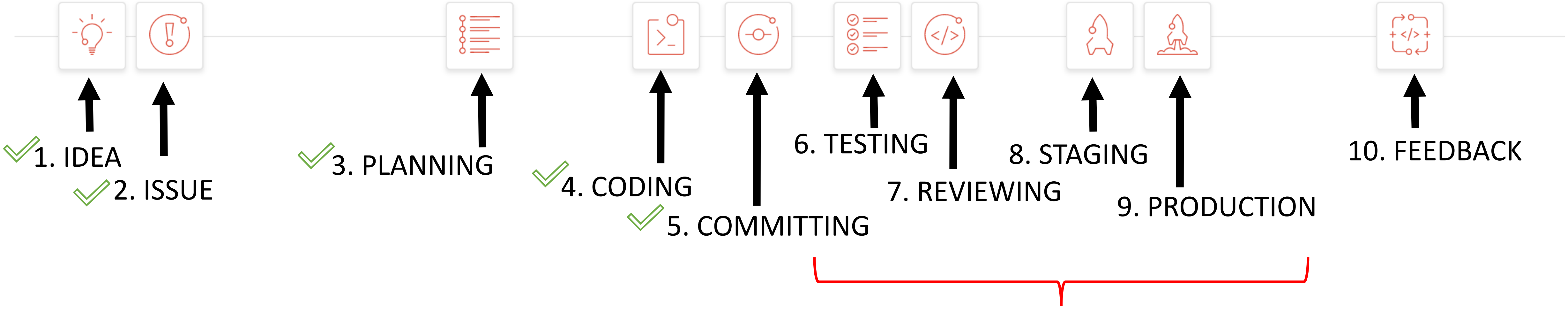
Automated testing

- We want each commit tested (“by trusted party”)
- We merge only if **pass tests!**

- **Continuous integration**

  each commit that we push to the origin is tested git → GitLab

The conversational development paradigm



✓ 1. IDEA

✓ 2. ISSUE

✓ 3. PLANNING

✓ 4. CODING

✓ 5. COMMITTING

6. TESTING

7. REVIEWING

8. STAGING

9. PRODUCTION

10. FEEDBACK

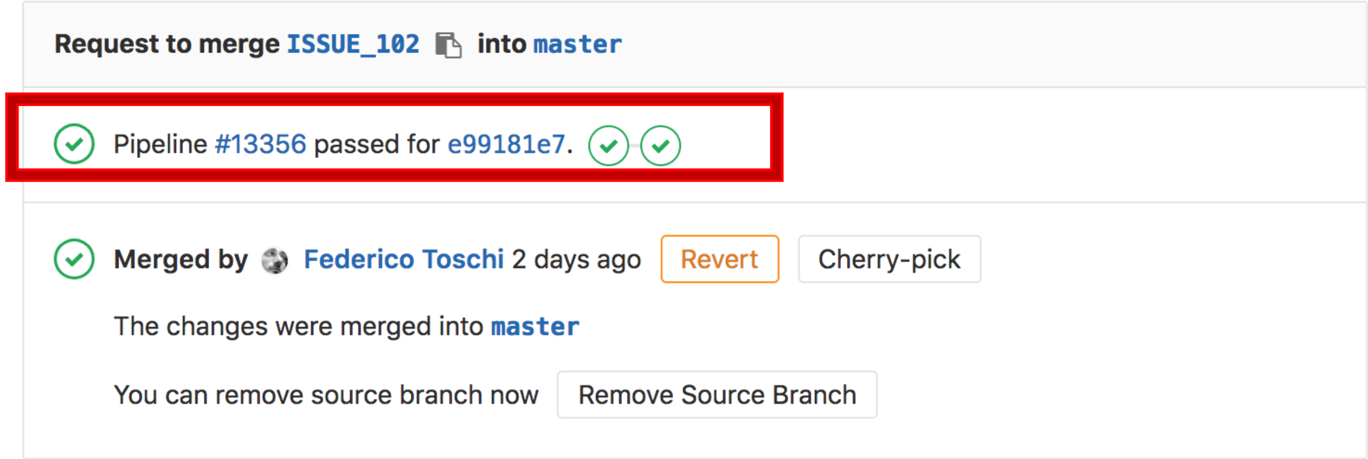
Today

Automated testing

- We want each commit tested (“by trusted party”)
- We merge only if **pass tests!**

- **Continuous integration**

each commit pushed to the origin is tested



Request to merge ISSUE_102 **into** master

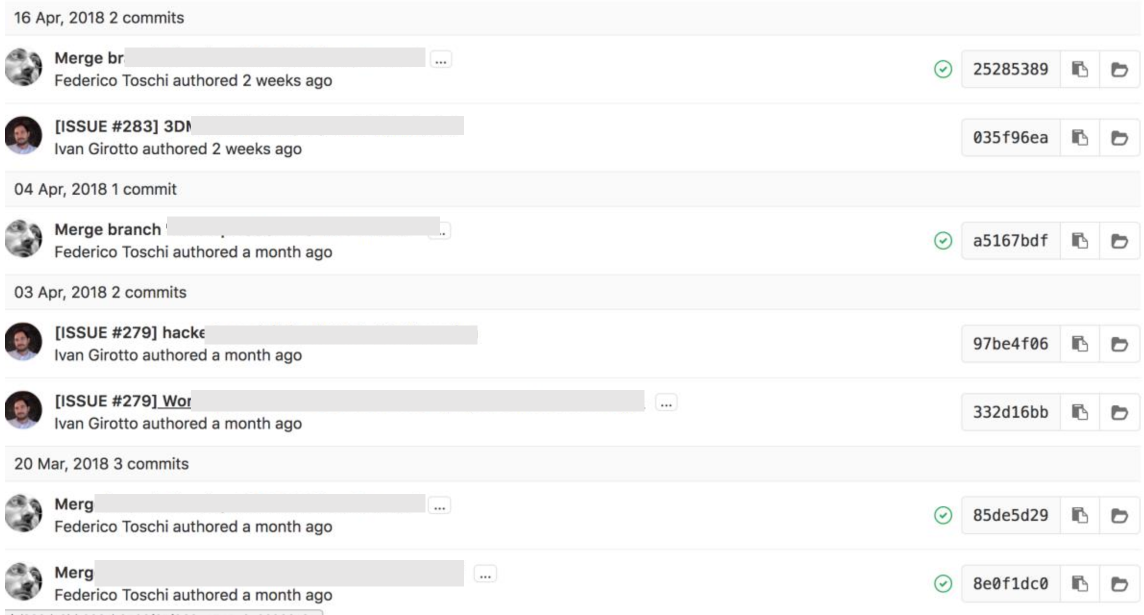
✓ Pipeline #13356 passed for e99181e7. ✓ ✓

✓ **Merged by** Federico Toschi 2 days ago

The changes were merged into **master**

You can remove source branch now

I can merge safely: tests are passed



16 Apr, 2018 2 commits

Merge br [redacted] ...
Federico Toschi authored 2 weeks ago ✓ 25285389

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

04 Apr, 2018 1 commit

Merge branch ' [redacted] ...
Federico Toschi authored a month ago ✓ a5167bdf

03 Apr, 2018 2 commits

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

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

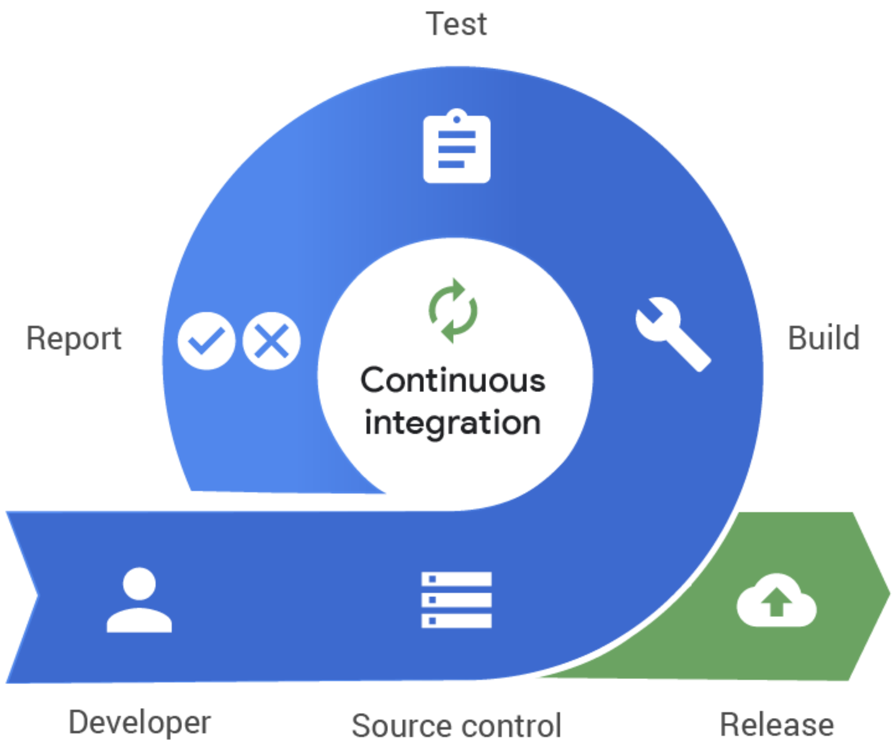
20 Mar, 2018 3 commits

Merg [redacted] ...
Federico Toschi authored a month ago ✓ 85de5d29

Merg [redacted] ...
Federico Toschi authored a month ago ✓ 8e0f1dc0

Continuous integration

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



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

Contemporary testing anatomy

We `ASSERT` that our software satisfies requirement

test passed if the assertion satisfied; fails otherwise

In general:

1 Test = piece of software that checks **1** `ASSERT`

Scientific method analogy

Scientific method:

We run experiments to invalidate
(or “prove/hinting correctness of”) a hypothesis.

A test is like an experiment

Functional testing paradigm:

Arrange → preparation of script checking feature

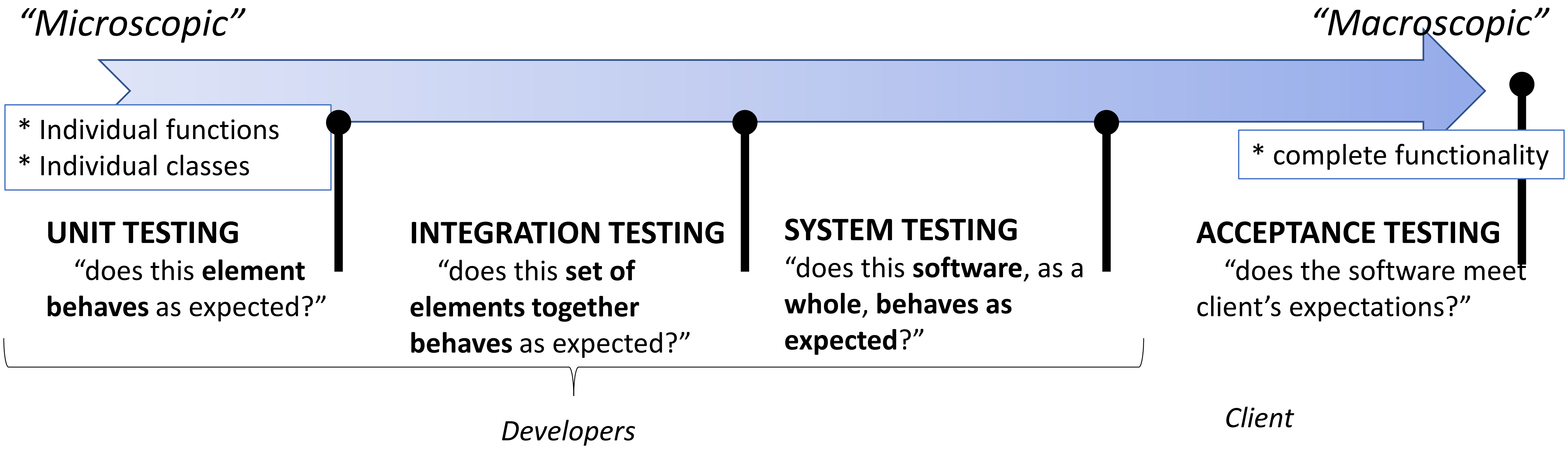
Act → run test

`ASSERT` → failure = issue

Scales of testing

Good modern software: collection of weakly coupled modules.

Testing follows these scales



“Microscopic”

“Macroscopic”

* Individual functions
* Individual classes

UNIT TESTING
“does this **element behaves** as expected?”

INTEGRATION TESTING
“does this **set of elements together behaves** as expected?”

SYSTEM TESTING
“does this **software**, as a **whole, behaves as expected**?”

* complete functionality

ACCEPTANCE TESTING
“does the software meet client’s expectations?”

Developers

Client

In other words

- **Unit test:** if fails -> a piece of code needs fixing.
- **Integration test:** if fails -> components not working *together properly*.
- **System test:** if fails -> the application not working as expected
- **Acceptance test:** if fails -> the application not doing what the customer expects.
- **Regression test:** if ails, the application no longer behaves the way it used to.

”Contemporary” vs. “traditional” testing

Issues emerge with **usage** or In **production**

- **Traditional approach & pitfalls**
  - System tested as a whole
  - High complexity (how do I build a proper test?)
  - Hard to test individual components
  - Hard to find sources of errors
  - Testing done through print statements/debuggers/script

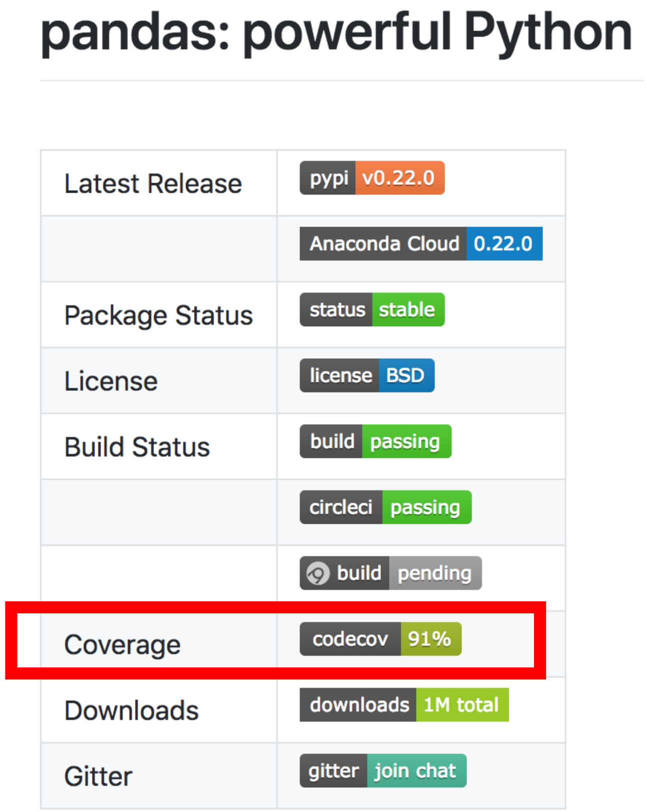
Issues emerge In **development**

- **Contemporary/Unit testing**
  - Lower complexity
  - Compliance to past requirements easily checked by module
  - To the limit: **Test Driven Development** (TDD – Exercise this afternoon)
    - As a requirement is identified, tests are written before the implementation

White box testing “quality” metrics: coverage

- If parts of our code are not tested (i.e. not covered by a test)
  bugs have higher chance to reach production

- **Line coverage**
  - **Percentage of lines** covered by at least 1 test
    (an if condition might be unsatisfied in all tests, thus the if-true branch remains always untested).
- **Branch coverage**
  - 100% line coverage might still leave many branches (that grow combinatorically) unexplored.



pandas: powerful Python

Latest Release	pypi v0.22.0
	Anaconda Cloud 0.22.0
Package Status	status stable
License	license BSD
Build Status	build passing
	circleci passing
	build pending
Coverage	codecov 91%
Downloads	downloads 1M total
Gitter	gitter join chat

Unit testing heuristics 1

1. Create test when object design is complete

   In TDD write test when interface is defined

2. **Design components that are testable**

   Make life of a tester easy: e.g. allow swappable mocks

3. Testing time slows down development

   make quick tests (at run time)
   make tests that are no-brainer to run

4. Develop tests using effective number of testing cases

   Heads up: generally combinatorial explosion of inputs, cannot be matched by as many `ASSERT`

   Selecting relevant & (all) edge cases -> more practice than theory

Unit testing heuristics 2

5. If possible compare e.g. with analytic solution or even slower-but-working versions of the same algorithms

   (Model-based testing)

6. In computing: knowing about internal computing mechanisms to make relevant tests
   1. **respect computer arithmetic**
   2. **Avoid non-determinism/fix seed in testing**

7. **REM: a failed test means a bug is introduced, not the other way around!**

8. **Best practice: Every new bug -> new test**

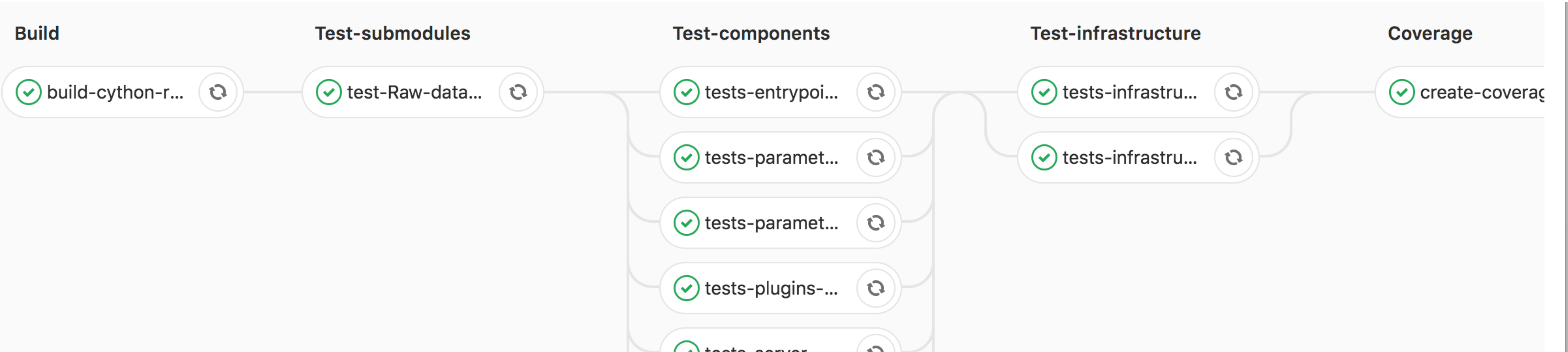
   against future regressions (e.g. from rollbacks)

Continuous integration in gitlab

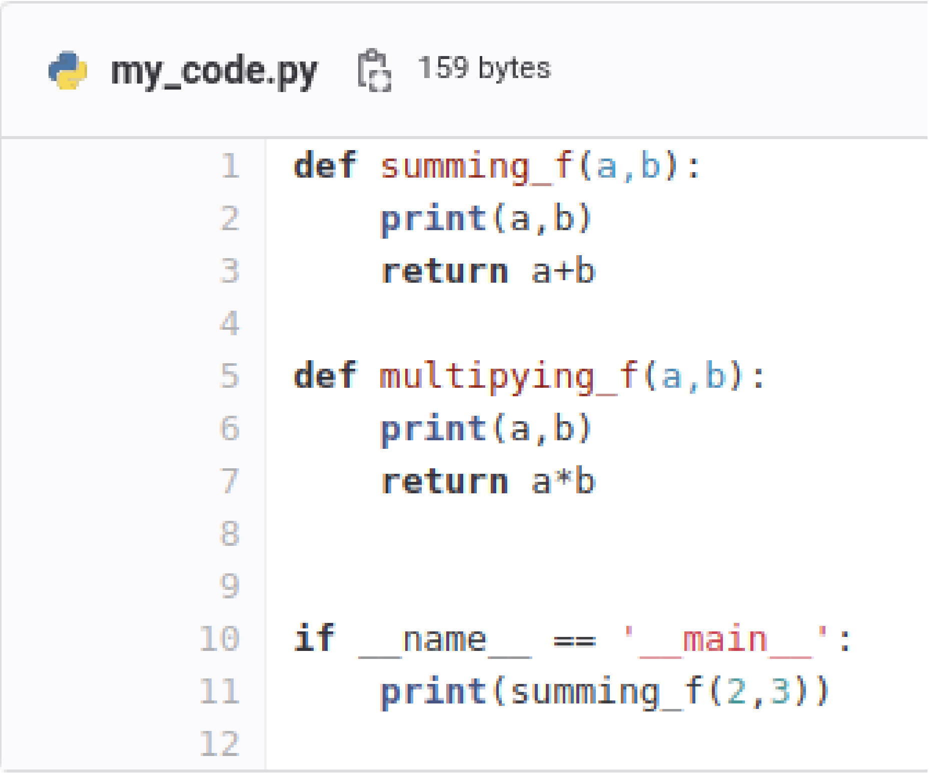
- a file in the repo root

  `.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.

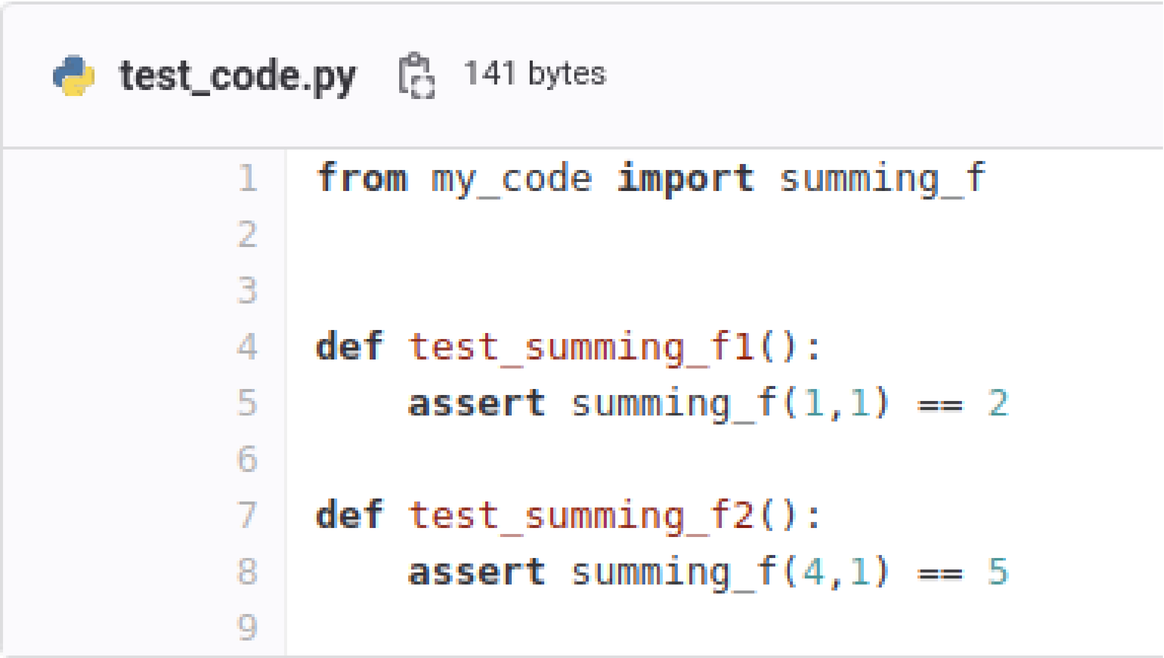


Minimal python example



my_code.py 159 bytes

```python
def summing_f(a,b):
    print(a,b)
    return a+b

def multipying_f(a,b):
    print(a,b)
    return a*b


if __name__ == '__main__':
    print(summing_f(2,3))

```



test_code.py 141 bytes

```python
from my_code import summing_f


def test_summing_f1():
    assert summing_f(1,1) == 2

def test_summing_f2():
    assert summing_f(4,1) == 5

```

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

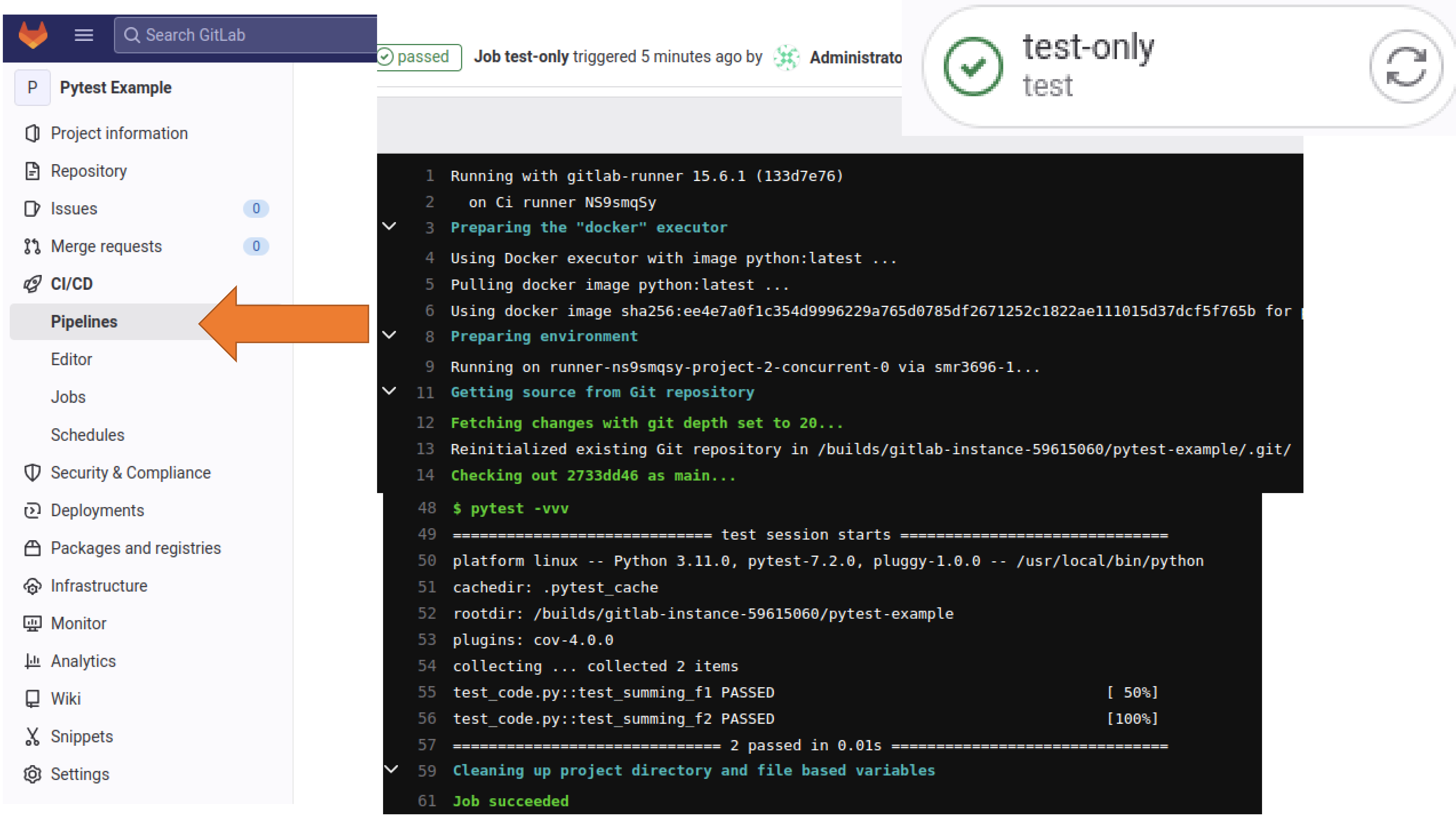
```
image: python:latest
```

Running in linux sandbox with up-to-date python
(Docker container, more in next lecture)

```
test-only:
   script:
      - pip install pytest
      - pytest -vvv
```

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

Testing



passed Job test-only triggered 5 minutes ago by Administrato

```
Running with gitlab-runner 15.6.1 (133d7e76)
  on Ci runner NS9smqSy
Preparing the "docker" executor
Using Docker executor with image python:latest ...
Pulling docker image python:latest ...
Using docker image sha256:ee4e7a0f1c354d9996229a765d0785df2671252c1822ae111015d37dcf5f765b for
Preparing environment
Running on runner-ns9smqsy-project-2-concurrent-0 via smr3696-1...
Getting source from Git repository
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /builds/gitlab-instance-59615060/pytest-example/.git/
Checking out 2733dd46 as main...
$ pytest -vvv
============================ test session starts =============================
platform linux -- Python 3.11.0, pytest-7.2.0, pluggy-1.0.0 -- /usr/local/bin/python
cachedir: .pytest_cache
rootdir: /builds/gitlab-instance-59615060/pytest-example
plugins: cov-4.0.0
collecting ... collected 2 items
test_code.py::test_summing_f1 PASSED                                     [ 50%]
test_code.py::test_summing_f2 PASSED                                     [100%]
============================= 2 passed in 0.01s ==============================
Cleaning up project directory and file based variables
Job succeeded
```

See examples

https://git.smr3696.ictp.it/gitlab-instance-59615060/**pytest-example**

Continous deployment

- One step further: serving the product immediately

- Example: static pages that serve **docs**, **executables**…

Minimal python example: testing

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

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

.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
```

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

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
```

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

Result is sent back to the server.