MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**fabien.coronis@maxiv.lu.se**) was recorded on submission of this form.

Email *

fabien.coronis@maxiv.lu.se

General

Which facility? *

MAX IV Laboratory, Lund, Sweden

Which beamlines? *

BioMAX and MicroMAX

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

MXCuBE-Web 3 at BioMAX (upgrade to 4 in progress).
MXCuBE-Web 4 at Micromax.

Resources

How much developer resources do you have?

How many people work on MXCuBE?

4-5

How many are software developers?

3-4

How many beamlines with MXCuBE do they have to take care of?

2

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

Maybe about 80h per week in total across all of us.

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

Limited for BioMAX.
Quite far for MicroMAX with help of an "emulator".

How often do you need to go directly at the beamline to code (not test)?

Always for BioMAX.
In many cases not necessary for MicroMAX.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

Ask questions to each other. Per email and during dev sprint meetings.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Keep an eye incoming pull requests.
Our regular process is to rebase our MAXIV specific changes on top of the latest GitHub upstream.

How often do you integrate upstream changes?

At best every second week. Sometimes can be 2-3 months between upstream integrations.

Do you cherry-pick specific changes?

For example important bug fixes only.

Can happen.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, GitLab, not public but should be.

How often do you contribute your improvements back to upstream GitHub?

Not as often as we would like.
We try to get a change on upstream first when the change in question is not critical for us.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

Good for the software developers.
Not so good for non-software developers.

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

MicroMAX: a lot can be tested with the "emulator".
BioMAX: everything has to be tested at the beamline.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

Yes for MicroMAX: the "emulator" is a docker-compose based reproduction of the beamline (not the whole beamline, the parts relevant to MXCuBE).

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

MicroMAX and future BioMAX: conda environment
BioMAX current: Docker containers

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

MicroMAX: We install some "hard to install" dependencies with conda. And the rest with pip, including mxcubeweb and mxcubecore that are installed as editable.

How do you install JavaScript dependencies?

MicroMAX: pnpm, nothing special

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

MicroMAX: Rock Linux 8.7. This is policy from MAXIV software team, we aim to have same OS everywhere, and the choice is Rocky Linux. We try to update the version regularly (once a year?). We have bunch of PXE servers for OS installation and Ansible playbooks for configuration of OS and installation of software within OS.

BioMAX: all done by hand, Docker container wit Portainer.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

The beamline network where we deploy MXCuBE is cut off from Internet.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

We have Ansible infrastructure and custom/local repositories (conda, PyPI, and npm).

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**baokangwen@gmail.com**) was recorded on submission of this form.

Email *

baokangwen@gmail.com

General

Which facility? *

SSRF

Which beamlines? *

BL19U1

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

Web ISPyB/EXI CBF ADXV

Resources

How much developer resources do you have?

How many people work on MXCuBE?

2

How many are software developers?

2

How many beamlines with MXCuBE do they have to take care of?

3

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

20h/week

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

pycharm

How often do you need to go directly at the beamline to code (not test)?

1day

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

Only read the code yourself, GPT assisted

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Participate in a code camp meeting

How often do you integrate upstream changes?

None

Do you cherry-pick specific changes?

For example important bug fixes only.

Some changes have been made and some bugs have been fixed

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes，We don't want to be apart just yet

How often do you contribute your improvements back to upstream GitHub?

None

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

I think that's pretty awesome

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

Offline or light testing

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

Epics

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Encountered some communication problems

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Conda

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

conda

How do you install JavaScript dependencies?

npm

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

centos7
local management

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

No restrictions

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

We have access
We don't use Kubernetes,Ansible and anything other deployment

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

My coding skills are relatively poor, and I find it difficult to understand.
I want to add a new content hardware, I don't know how to get started, for example, I want to add a robot to the MXCUBE, I don't know how to do it, the code call is too complicated.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Sorry, I'm not at the level of being able to provide advice at the moment.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**r.h.fogh@gmail.com**) was recorded on submission of this form.

Email *

r.h.fogh@gmail.com

General

Which facility? *

Global Phasing

Which beamlines? *

None

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

All of the above

Resources

How much developer resources do you have?

How many people work on MXCuBE?

1

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

currently 6 are integrating the workflow

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

20 hrs / week

Do you have any additional comments?

We are not a synchrotron

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

100%

How often do you need to go directly at the beamline to code (not test)?

never

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

No. Queue structure, standards, abstract classes need talking / documentation

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Fetch and rebase

How often do you integrate upstream changes?

every week or two

Do you cherry-pick specific changes?

For example important bug fixes only.

No

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes. www.github.com/rhfogh, public

How often do you contribute your improvements back to upstream GitHub?

Whenever a change is nready. Monthly or better

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

Hooks: 0%. Autotest 0%, lint, black, github (all in Pycharm) 95%

Do you have any additional comments?

Autotest has NEVER passed one of my pushes. I ignore it.

Test

How do you test new MXCuBE code?

How do you test new code?

Running MXCuBE in mock mode with image nsimulation thatp produces actual data

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

No, only mock objects

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Mock objects are sometimes not kept up not date by others.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Conda for our own testing. Deployment depends on synchrotron

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

Conda, poetry, pip.

How do you install JavaScript dependencies?

Full poetry reinstall for new block of work; maybe monthly.

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Test on Linux, OpenSUSE - Full management freedom. Deployment depends on synchrotrons

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

No limits

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

For tdting am in full control; no complex tools beyond conda, pypi, std. install sctipt tools

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Autotest never has worked. Hard to keep refactoring for multiple, parallel PRs (unavoidable). Looking forward to time when all synchrotrons have caught up with upstream develop - or there was a stable, widely used version pone could program and develop to (unlikley to happen).

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Get autotest to work for me. Continue code cleanup and harmonisation.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**andrey.gruzinov@desy.de**) was recorded on submission of this form.

Email *

andrey.gruzinov@desy.de

General

Which facility? *

DESY (Hamburg, Germany)

Which beamlines? *

P11

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

PyQt5, mxcubeqt, ISPyB/EXI (in development), HDF5 (1000 frames/file), Albula (via python bindings)/ adxv (via custom listener), autoprocessing on the SLURM-based cluster vial slurm jobs per ssh.

Resources

How much developer resources do you have?

How many people work on MXCuBE?

1

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

1

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

50-70% for "offline" improvemnents (~3-5 full days per month for testing with the beam) (in active development phase at P11, try to bring it into production ASAP), otherwise dependent on the other beamline tasks.

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

There are only cosmetic code improvements (linting, etc.) due to the network access limitations for Tango servers (too many errors due to the inability to load certain components). Most of the development happens via SSH login on the target machine.

How often do you need to go directly at the beamline to code (not test)?

In principal access to the beamline is needed only for testing.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

Mostly code reading and direct emails. Since the documentation became available - documentation too.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

git pull + git rebase from upsteream develop before the changes to the local branch

How often do you integrate upstream changes?

DUe to the active development I'm trying to do it as frequently as possible (~every week)

Do you cherry-pick specific changes?

For example important bug fixes only.

Only if some major breaking happens, mostly rebasing to the latest develop and direct debug on my side.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, on agruzinov on GitHub, but currently I've adapted usage of the main upstream. Using fork under my account was causing some problms with rebasing.

How often do you contribute your improvements back to upstream GitHub?

if it is small PR, immediately as the bug on my side is found. If it is some major improvements for the site specific code, probably once per month open the PR and keep it in WIP mode until the it is logically finished.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

Using on the daily basis

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

Directly on the beamline if possible.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

No, but it would be interesting to know if someone is doing that.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

N/A

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Standard Python venv, docker is not recommened, podman as alternative may be used.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

initial install using poetry in venv, than using pip.

How do you install JavaScript dependencies?

N/A

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Linux Debian 10/12 (local policy, imposed by guidlines with centralised management and no sudo rights).

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

Anaconda can not be used; mamba is an alternative, but it is not used for MXCuBE development in particular—settled with venv. Local Maschine Python is restricted to the changes—a separate Python installation compiled from the source.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

None that are used for MXCuBE. Kubernetes is generally avalable through central IT.

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

In the current active developmant phase access to the hardware is generally needed to fully test the behaviour. Simulated environment maybe a solution, but some particular bugs can be seen only in "production" (like network delays).

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

More documerntation (if possible), in particular on the Qt part and core-Qt communication. Maybe have some VM or docker container for the mock-up version for example to test compatibility when migrating to another os/environment/python version?

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**dan-tiberiu.costin@synchrotron-soleil.fr**) was recorded on submission of this form.

Email *

dan-tiberiu.costin@synchrotron-soleil.fr

General

Which facility? *

SOLEIL

Which beamlines? *

Proxima 1

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

Qt v4.8.5 et MXCUBE 2.3 , python 2.7.5 en production avec MXCUBE wed en developement + ISPyB. Real time image viewer ADXV , auto processing XDSME + Autoproc.

Resources

How much developer resources do you have?

How many people work on MXCuBE?

1 full time developer (CDD ending in Mars 2024)

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

1

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

Full time

Do you have any additional comments?

Developer contract is for 18 moths

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

Not far. Tests are done on the beam line directly, so beam line availability is important. This is important not only for testing but also for learning about how the MXCuBE app behaves an how it works or brakes.

How often do you need to go directly at the beamline to code (not test)?

There is no direct coding on the production version of MXCuBE and coding for the development of the new MXCuBE web is inseparable from testing.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

I learn about the inner workings of MXCuBE through reverse engineering, exploring the core and testing. Reading the code is not enough, I thing testing and exploring the code live using tools like pdb or print outs is an important part of the learning process especially since deprecated classes and methods have not been removed and much of the current code base isn't in use.

Do you have any additional comments?

I like the documentation initiative. I think having a solid documentation base is vital to the development of the MXCuBE project especially considering that in the current state the code base has seen very little pruning or refactoring. As new code is added on top of old deprecated code it is becoming more time consuming for new developers to learn about the inner workings of the application. Robust documentation can significantly boost productivity.

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

I follow developments on Github and do a Git merge about one a month or whenever necessary, to update my development branch.

How often do you integrate upstream changes?

About once a month.

Do you cherry-pick specific changes?

For example important bug fixes only.

So far no. I keep my develop branch up to date with the upstream and merge it into my "feature" branch where I develop. I integrate the upstream changes like this about once or twice a month.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, I have my own forks of the Mxcubeweb, Mxcubecore, and video streamer repositories. These are public on Github.

How often do you contribute your improvements back to upstream GitHub?

Not as often as I'd like to. I only had about 2 or 3 pull requests in the first 6 month of my contract.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

I'm not super confident but I'm ok with it.

Do you have any additional comments?

It would be nice to find a way to increase inter developer interactions .

Test

How do you test new MXCuBE code?

How do you test new code?

I use my "feature" branch of the MXCuBE where I add new code, I run the app and see how it behaves. I use pdb and printouts for exploration.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

No simulated equipment.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Proxima 1 has very specific ways to handle the sample changer and goniometer. Development is faster using the real hardware objects.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

I use conda fro python, with the mxcubeweb envirnment. No docker containers.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

Mostly conda for mxcubecore and poetry for mxcubeweb.

How do you install JavaScript dependencies?

Poetry install on mxcubeweb .

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Linux Ubuntu 24.10

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

There are some network restrictions but we managed to get access to conda channels.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

No, so far nothing specific or custom for the deployment.

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

For me personally a recurring pain point is getting stuck because of missing knowledge about the system. Exploring the code to gain that information takes up most of my time.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

I think interactive coding sessions between small groups of 2-3 developers would boost learning and productivity.

Do you have any additional comments?

I think once yaml configuration is done, there should be some focus on cleaning and refactoring the code to remove old unused classes methods and attributes and also make the code more secure, efficient and clear.

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**lais.pessine@ess.eu**) was recorded on submission of this form.

Email *

lais.pessine@ess.eu

General

Which facility? *

ESS (European Spallation Source), located in Lund, Sweden.

Which beamlines? *

NMX

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

- MXCuBE: Web;
- Python: 3.10;
- LIMS: SciCat;
- File format: HDF5/Nexus;
- Real-time image viewer: TBD (will need to be compatible with new NMX detector);
- Autoprocessing workflows: SCIPP/ESSNMX (https://github.com/scipp/essnmx), DIALS;

- Other UI frameworks: NICOS (nicos-controls.org). NICOS is aimed at experiment control, like MXCuBE, but it is general purpose, and the default UI for ESS instruments. At NMX, the plan is to have MXCuBE as the main UI, and NICOS in parallel for prototyping, calibration, expert view, and backup.

Resources

How much developer resources do you have?

How many people work on MXCuBE?

2 people (Aaron Finke and Lais Pessine).

How many are software developers?

1 person (Lais).

How many beamlines with MXCuBE do they have to take care of?

1 beamline (NMX).

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

Around 16 hours (2 working days) per week. If there aren't other demands in the week, this time can be increased.

Do you have any additional comments?

At ESS, we are still in the beginning of our MXCuBE journey with neutrons: NMX beamline being installed, and the facility finishing its construction phase. Thus, the times and some practices described in this form may vary in the upcoming months.

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

For now, the work is done with a local computer (laptop) and lab hardware, as the beamline devices and its infrastructure are still being delivered/installed.

Anyway, the goal is to have mockups and/or simulated hardware to test as much as possible the initial ideas locally, before using trying them on the beamline.

How often do you need to go directly at the beamline to code (not test)?

For now, once every 1 or 2 weeks is expected.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

I learn mainly from following the MXCuBE tutorials, reading the code, documentation of some of its dependencies (some of them linked in MXCuBE docs), and MXCuBE code camps.

For doubts not solved this way, the MXCuBE community is reached for search or discussion (through the MXCuBE Github threads, mailing list, developers meetings, or in-person meetings).

Do you have any additional comments?

No.

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Enabling notifications for the Github upstream repo, and through the mailing list.

How often do you integrate upstream changes?

As we are in a starting phase, we try them locally and integrate them in production as soon as possible, using the latest tag/version available.

Do you cherry-pick specific changes?

For example important bug fixes only.

We didn't have the need of cherry-pick yet. But would be okay to do this to fix something temporary.

The preferred way would be, however, to test and rebase with the next tag/version containing the fix, as soon as we have time for it.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

We have our public fork at Github, for sharing and contributions:
* https://github.com/ess-dmsc/mxcubeweb
* https://github.com/mxcube/mxcubecore

and also in our internal ESS GitLab.

Although we will work with Neutrons MX, the plan is to avoid diverging from the upstream within these forks, keeping the same base code as much as possible, just with different configuration.

How often do you contribute your improvements back to upstream GitHub?

At the last official Meeting (May 2024, at MAX IV) we contributed with some small fixes in documentation. But the plan is contribute back with fixes and features whenever possible.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

We feel comfortable using them, also because here at ESS we use similar tools for other projects. In general, it has been good to have them for automation and keeping the code in good shape.

Do you have any additional comments?

No.

Test

How do you test new MXCuBE code?

How do you test new code?

With simulated hardware, and hardware in the lab (usually in this order, but also depending on what is available).

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

Yes. EPICS simulated IOCs. They can be written with EPICS itself, or with Python libraries.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

With motors, for example, the production IOC or device may have more parameters to be considered and calibrated than its simulated version, which can lead to unexpected behaviors when testing with real hardware after.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Yes, we use it. Currently, we are using Conda and Poetry (as suggested by mxcube-web tutorials). So far, it's working fine. But the plan is to try to remove Poetry from the production environment, to keep just Conda or Venv.

Docker containers are not being used now.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

With Poetry (again following what is suggested by mxcube-web tutorials), and in some cases with conda. So far, it's working fine. But as said before, the plan is to try to remove Poetry from production.

How do you install JavaScript dependencies?

With pnpm.

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

This part is still under validation, but we are currently deploying on Linux Ubuntu 22.04, in a virtual machine. We are likely to use Ubuntu 22.04, AlmaLinux 8 or AlmaLinux 9 since we are currently migrating from CentOS 7. The management is local.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

We have some licensing restrictions in place. In general we can use any open-source software, but use of software that requires modification tends to be limited to the MIT license.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

Yes, we have an ESS PyPI repository. Ansible is also used for deployment.

Do you have any additional comments?

No.

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

For now, no real pain points. But the main challenge is to prepare/develop/test mxcube for the neutrons use case, trying to keep up with the upstream code.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Surveys like this one (and the one about cyber security) are good initiatives, and could be done more often for knowledge sharing and discussion in the meetings.

Do you have any additional comments?

No.

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**nicolas.molitenro@lnls.br**) was recorded on submission of this form.

Email *

nicolas.molitenro@lnls.br

General

Which facility? *

LNLS (Sirius)

Which beamlines? *

MANACÁ

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

Web, no LIMS, CBF, ADXV image viewer, in-house XDS-based autoprocessing pipeline (MANACAutoProc), web-based tools (puck naming, experimental station cameras)

Resources

How much developer resources do you have?

How many people work on MXCuBE?

2

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

1

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

+- 14 hours per week

Do you have any additional comments?

At LNLS, as only the manacá beamline currently uses MXCuBE, we develop the codes during approximately Monday to Wednesday and on the same days testing in a production environment. Because Monday and generally Tuesday are used for machine study. The other days the line meets with users. And with that we work on other demands.

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

By coding only on my laptop, I can move forward with the codes and tests. Occasionally, when beamline time is needed, it is discussed with the line users. But with the EPICS equipment that I simulate, we can move forward with developments.

How often do you need to go directly at the beamline to code (not test)?

The days when I can go to the light line to develop are usually Mondays and some Tuesdays. Because these are the days for machine study. Generally when it is necessary to develop a piece of equipment and have to be present with it.

How do you learn about the inner workings of MXCuBE?
Is it enough to read the code?

Reading the code documentation helps to understand. But develop the equipment classes with tests.

Do you have any additional comments?

I think Codecamp and dev meetings are great for the group's knowledge. And they helped us understand the project a little better.

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

With dev meetings and always looking at the project repository, if there is an issue or any change in the project.

How often do you integrate upstream changes?

We use gitLab here at LNLS, which is currently closed to internal users only. I spoke to Marcus to include me in the MXCuBE git. We are updating and testing to migrate to version 4 definitively. With that, access to git would help us share our developments.

Do you cherry-pick specific changes?

For example important bug fixes only.

Yes

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, I'm using the gitlab, this forks are private.

How often do you contribute your improvements back to upstream GitHub?

Rarely, but I would like to share more, as soon as my access is released I intend to update some of our EPICS classes, as well as help maintain the entire project.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

have knowledge of the tools, and I use them on a daily basis, but I am more familiar with git.

Do you have any additional comments?

No

Test

How do you test new MXCuBE code?

How do you test new code?

We test the code on our MXCuBE dev, using simulated EPICS equipment. When necessary, we test in production with real equipment and user conditions.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

Yes, EPICS

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

The mockup hardware will always be a static piece of equipment, simulating the mockup as a real one is difficult, as it "always works", and depending on how the real behavior of the equipment is configured, the mockup does not always resemble it.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Yes, using conda, micromamba and docker containers.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

Install with conda.

How do you install JavaScript dependencies?

npm install

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

I'm using Linux, my PC is a debian 12. MXCuBE project pc is a debian 10.
Updates packages localy.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

The only IT restriction policy is the network, mainly with VPN.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

Yes. No not have custom conda or npm and not use Kubernetes.

Do you have any additional comments?

No

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Most of the time it is necessary to include a new device, mainly depending on two states, when values are set.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

More talk to two developers, and help the team. As there are different ways to control the LABS equipment through software other than codeCamp, with examples simulated by tango and EPICS will help you better understand the project and the new developments for the project.

Do you have any additional comments?

No

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**liao.zeno@nsrrc.org.tw**) was recorded on submission of this form.

Email *

liao.zeno@nsrrc.org.tw

General

Which facility? *

NSRRC

Which beamlines? *

TPS 15A

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

Web, no LIMS, HDF5(Eiger detector), Albula

Resources

How much developer resources do you have?

How many people work on MXCuBE?

1

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

1 (actually, under construction right now)

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

few hours a week, I also working on construction of other beamlines

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

How often do you need to go directly at the beamline to code (not test)?

Not often because it is on construction now

How do you learn about the inner workings of MXCuBE?
Is it enough to read the code?

I am still not familiar with them, it would be help if there are some tutorial to describe the "inner workings" parts of MXCuBE.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

How often do you integrate upstream changes?

Do you cherry-pick specific changes?

For example important bug fixes only.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

How often do you contribute your improvements back to upstream GitHub?

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

I didn't use these tools before, but I would like to try.

Do you have any additional comments?

Because it is under construction, I did not try to do these things now.

Test

How do you test new MXCuBE code?

How do you test new code?

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

I am try to use conda now, no docker

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

conda

How do you install JavaScript dependencies?

conda? or what should I notice?

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Ubuntu

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

no, I can use Anaconda

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

no

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**oscarsso@esrf.fr**) was recorded on submission of this form.

Email *

oscarsso@esrf.fr

General

Which facility? *

ESRF

Which beamlines? *

ID23-1, ID23-2, ID30A1, ID30A3, ID30B, ID29 and BM07

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

- MXCuBE-WEB,
- ISPyB and ICAT
- HDF5, CBF
- ADXV, Braggy and Albula
- BES Workflows
- GPHL Workflows
- BLISS control system

Resources

How much developer resources do you have?

How many people work on MXCuBE?

2 (3 for the next year(s)) + 50% from EMBL Grenoble and we recently had additional ~20% FTE for web development.

How many are software developers?

All are software developers or have engineering background

How many beamlines with MXCuBE do they have to take care of?

7.5 beamlines and 3 other instruments taken care of by 2 (now 3 people)

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

10-20h per person and per week including collaboration and code review

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

Having a mock-up/demo version is useful in order to be able to get an idea whether a feature or solution works or not, or to check for regressions.

How often do you need to go directly at the beamline to code (not test)?

Every week - for verification and validation, the code is mostly written locally

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

The collaboration is of great help and there is always someone that can answer the questions we have. We otherwise, get fairly far by looking at the code and trying to understand how things are working.

Reading the documentation and contributing guidelines also helps in the beginning. The code camps are also very useful for leaning about mxcube, methodology and best practices.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Non ESRF specific work is done directly towards GitHub which is re-based on the internal GitLab repository on a weekly basis

How often do you integrate upstream changes?

Several times per week

Do you cherry-pick specific changes?

For example important bug fixes only.

No

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

GitLab but we would like to move those to GitHub

How often do you contribute your improvements back to upstream GitHub?

Several times per week

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

We feel comfortable and confident

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

All changes (not only the ones made at ESRF) are tested on a beamline on a machine day before deployment.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

No

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

That the mockup rarely behave as real hardware. Difficult to reproduce the functionality you would like, issues with timing, protocols

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Yes, conda

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

poetry and pip

How do you install JavaScript dependencies?

pnpm

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Ubuntu 20.04, Python 3.10.x

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

Infrastructure and IT team restrict access to certain services and filesystems. Access to PyPi, npm registry and conda channels are open

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

We use Ansible and GitLab for certain software

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Testing and deployment are two topics that are quite difficult to handle. Tests are particularly difficult to perform as the hardware is difficult to simulate and test benches are expensive. Mockups are usually very general and not specific enough.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

- Testing could be improved in general
- Improving documentation
- Cleaning up the code base and removing deprecated and confusing code
- Task allocation within the collboration

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**oldfielj@ansto.gov.au**) was recorded on submission of this form.

Email *

oldfielj@ansto.gov.au

General

Which facility? *

Australian Synchrotron - ANSTO

Which beamlines? *

MX3

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

Web/Qt: MXCuBE-Web
LIMS: MX-Prism
File Format: HDF5
Real-Time Image Viewer: N/A
Autoprocessing Workflows: Prefect Based - MX3 UDC (FastDP, Dials, Best)

Resources

How much developer resources do you have?

How many people work on MXCuBE?

5

How many are software developers?

2

How many beamlines with MXCuBE do they have to take care of?

1

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

0-80hr/s Per Week

Do you have any additional comments?

Work related to MXCuBE is currently limited, MX3 is currently in hot commissioning.

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

MXCuBE-Web and MXCuBE-Core running locally in containers, using simulated Ophyd devices.

How often do you need to go directly at the beamline to code (not test)?

Frequently to test hardware object implementations.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

Trial and error combined with extensive use of debugging tools.

It'd be nice if there was technical documentation to assist in implementing hardware objects for each abstract hardware object.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

We maintain an ANSTO fork on GitHub and merge upstream changes in as we are able.

How often do you integrate upstream changes?

Currently, infrequently, we're six months out of date.

Do you cherry-pick specific changes?

For example important bug fixes only.

No, we merge all changes from upstream when we do merge.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, on GitHub.

How often do you contribute your improvements back to upstream GitHub?

Currently, infrequently, should become more frequent once we're through hot commissioning.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

Very, we use these tools as standard on all our internal repositories.

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

Either using simulated Ophyd devices if possible, otherwise we do so by testing against beamline hardware.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

Yes, simulated Ophyd devices.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Differences in code behavior, e.g. hardware issues which are hard to emulate in simulated mode.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

We use Docker containers, MiniConda, and Python virtual environments.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

Poetry is preferred, if it's not PIP installable we default to Conda.

How do you install JavaScript dependencies?

NPM Build

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Linux/Debian-12, choice of distro is left up to us, packages left to developers to maintain and update, manual process.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

Yes, we use Kubernetes deployments and must integrate with centralised single single sign on service, we maintain an internal PyPi cache.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

Internal PyPi repositories, Kubernetes for deployment, Drone for CI, ArgoCD for CD.

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Merge conflicts, limited test cases.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Additional test cases for abstract hardware objects, example implementations of abstract hardware objects with developer documentation.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**alessandro.olivo@elettra.eu**) was recorded on submission of this form.

Email *

alessandro.olivo@elettra.eu

General

Which facility? *

ELETTRA

Which beamlines? *

XRD1, XRD2

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

XRD1:
- MXCuBE: Web version 4.27 (+ local changes)
- Control system: TANGO
- LIMS: ISPyB / VUO
- File format: CBF, TIF
- Viewer: Albula, Synchweb
- Auto-processing: Not yet implemented

XRD2:
- MXCuBE: Web version 3.0.0 (+ local changes)
- Control system: TANGO
- LIMS: ISPyB / VUO
- File format: CBF
- Viewer: Braggy, Synchweb, Albula,
- Auto-processing workflow: At the end of a data collection fastdp and autoproc pipelines are executed on two nodes using a tool developed at Elettra which allows starting scripts remotely via http (fastapi based). The OUTPUT FILES produced can be browsed through Synchweb

Resources

How much developer resources do you have?

How many people work on MXCuBE?

2 (+1 see comments)

How many are software developers?

2

How many beamlines with MXCuBE do they have to take care of?

2

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

On average, 24 per week.

Do you have any additional comments?

The people are:
1) A beamline Scientist: Currently contributing to the development of MXCuBE version 3.0.0.
2) A senior Developer: Played a key role in the initial development of MXCuBE version 3.0.0 and continues to provide ongoing support occasionally (XRD1).
3) A developer: Actively participating in the ongoing development of MXCuBE version 3.0.0 and its upcoming upgrade to the version 4 (XRD1), developing MXCuBE version 4 (XRD2).

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

A large part of the code can be developed on the personal laptop, either by interfacing with the beamline control system or using HWO mockups.

How often do you need to go directly at the beamline to code (not test)?

Development on the beamline is done for safety reasons (e.g., motors that might crash), to quickly solve problems that arise during beamtime, or to address more complex issues that also involve the specific boundary conditions of the environment where the software runs.

How do you learn about the inner workings of MXCuBE?

Is it enough to read the code?

Mainly reading the code and following the MXCuBE codecamp.

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

We synchronize the forked repository on GitHub, perform a fetch locally, and then propagate the changes to the repository stored on Elettra GitLab.

How often do you integrate upstream changes?

Initially, we integrated all changes approximately every three months. However, we have now decided to maintain a fixed version for one or more beamtimes, with the possibility of supporting and fixing issues, as well as adding new features requested by beamline scientists; but in the near future, we aim to adopt a workflow based on more frequent updates.

Do you cherry-pick specific changes?

For example important bug fixes only.

No, but it seems a good practice we could follow in the future.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, we have a forked github repository publicly accessible:
- mxcubecore: https://github.com/ELETTRA-SincrotroneTrieste/mxcubecore
- mxcubeweb: https://github.com/ELETTRA-SincrotroneTrieste/mxcubeweb

How often do you contribute your improvements back to upstream GitHub?

We haven't contributed yet, but we are following the contributing guidelines regarding branches and commits to be ready to submit a PR as soon as possible.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

We are becoming increasingly confident with Git and GitHub, also thanks to the project itself. Regarding the other tools, we still need to familiarize with them.

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

It depends on the type of test and the part of the code involved. Generally, since development mainly occurs on the personal laptop, the initial testing is done on this machine. Changes are then pushed to the repository stored on the Elettra GitLab and subsequently pulled to the test machine. At this point, the beamline scientists test and validate the changes.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

We use both generic Tango test devices, with which the HWOs interface, and the actual production Tango devices if they have a simulated operating mode. This is very useful for making MXCuBE's behavior more closely aligned with real-world scenarios.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Often, the most subtle problems occur due to interactions with the actual hardware

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

On the personal laptop, the deployment is done using Conda (following the MXCuBE installation guide), while on the test and production machines, the standard Python virtual environment (i.e. virtualenv) is used. Docker is not currently being used.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

We use pip on the test and production machines, while on the personal laptop, Poetry is used as recommended by the MXCuBE installation guide. However, for additional python libraries, we use pip.

How do you install JavaScript dependencies?

We are using npm.

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

We are using Rocky Linux 9 for test and production machines, a decision made by the sysadmin group, while Ubuntu 22.04 is used for the personal laptop, where the choice is more flexible. In both cases, however, the selections must comply with the Elettra policy. Management is centralized.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

We follow the internal Elettra policy for the deployment. There are no license restrictions for the deployment of MXCuBE. Regarding the network, we follow the internal Elettra policy.
We can technically use Anaconda, but for production environment it is discouraged. We have access to the package repository, but it's not needed for MXCuBE.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

We are not using our custom repository for MXCuBE. For infrastructure services, we use a service called RunScriptWeb, which allows remote processing of the collected data.

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Integrate upstream changes, change the part of the code related to queues.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.

MXCuBE development workflows

Questionnaire about development workflows for MXCuBE: coding, testing, deployment, collaborating, etc.

It would be interesting if each facility could share how they handle those topics. This way we can compare experiences, learn from each other, identify common pain points, and collaborate better on MXCuBE.

All questions (besides the first two initial questions) are optional, feel free to skip any question.

All questions expect free text answers, feel free to expand on some questions if you feel it is necessary.

Each section has an "*additional comments*" question, feel free to use it if the other questions do not correspond to what you want to write.

The respondent's email (**michael.hellmig@helmholtz-berlin.de**) was recorded on submission of this form.

Email *

michael.hellmig@helmholtz-berlin.de

General

Which facility? *

HZB-BESSY

Which beamlines? *

BESSY-MX 14.1, 14.2, 14.3

What is the software stack?

Web or Qt (PyQt version, etc.), LIMS (ISPyB/EXI, ICAT, etc.), file formats (HDF5, CBF, etc.), real-time image viewer (Albula, Braggy, etc.), autoprocessing workflows, and so on

MXCuBE-Qt (PyQt 4.x), no LIMS (currently in testing), CBF (Pilatus all beamlines), adxv/albula, XDSAPP autoprocessing on all datasets

Resources

How much developer resources do you have?

How many people work on MXCuBE?

1

How many are software developers?

1

How many beamlines with MXCuBE do they have to take care of?

3

How many hours can they dedicate to working on MXCuBE?

In total. Per week, month or year.

Any work stopped on MXCuBE after the cyber attack. Planned ~1/2 FTE during adaption to mxcubecore/mxcubeweb, 1d/week for maintenance.

Do you have any additional comments?

Coding

How do you code for MXCuBE?

How far can you get coding locally using your computer or laptop only?

Setup of development environment, testing with mockup version,

How often do you need to go directly at the beamline to code (not test)?

Regularly, testing of exact communication protocols of hardware devices.

How do you learn about the inner workings of MXCuBE?
Is it enough to read the code?

Started to work on MXCuBE when only the code was available. :-)

Do you have any additional comments?

Collaboration

How do you collaborate on MXCuBE code?

How do you keep up to date with upstream changes on GitHub?

Legacy Qt version in production, no upstream changes.

How often do you integrate upstream changes?

Legacy Qt version in production, no upstream changes.

Do you cherry-pick specific changes?

For example important bug fixes only.

Do you have your own forks of MXCuBE repositories?

If, yes do you use GitHub, GitLab or anything else like this? Are these forks publicly accessible?

Yes, fork on Github, public access.

How often do you contribute your improvements back to upstream GitHub?

Legacy Qt version in production, no contributions to upstream.

How confident do you feel using the collaboration tools?

For example: git and GitHub, pre-commit hooks, linting tools, formatters, and so on.

Generally use git and Github only.

Do you have any additional comments?

Test

How do you test new MXCuBE code?

How do you test new code?

Manual testing at the beamline with real hardware.

Do you use simulated devices (Tango, Epics, etc.) to test hardware objects?

No.

What limitations or hurdles do you encounter when testing with mockup hardware?

Compared to real hardware.

Generally mockup objects do not provide dynamic behaviour.

Deployment

How do you deploy new MXCuBE code?

Do you use isolation environments for Python?

Do you use conda environment?
Do you standard Python environment (venv, virtualenv, or Poetry)?
Do you use Docker containers?

Conda.

How do you install Python dependencies?

Do you install Python dependencies with conda, Poetry, pip?

Legacy Qt environment with pip, foreseen solution Conda.

How do you install JavaScript dependencies?

What operating systems do you deploy on?

Linux or something else? Which distribution? Which version?
Free choice or imposed by guidelines?
Centralised or local management (update cycle/package installation)?

Debian Linux 12+, AlmaLinux 8/9 with local management.

Do you have infrastructure restrictions for deployment?

Does you infrastructure/IT team restrict what you can use? Policies?
License restrictions?
Network restrictions?
Can you use Anaconda?
Do you have access to conda channels (package repositories), PyPI, npmjs.com?

No restrictions by IT.

Do you have access to specific infrastructure services?

Do you have your own custom conda, PyPI, npm repositories?
Do you use Kubernetes, Ansible, or anything like that for deployment?

No self-hosted repositories, no deployment tools.

Do you have any additional comments?

Pain points

What are recurring pain points regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

What do you think could be improved regarding MXCuBE development workflows?

In terms of coding, testing, deploying, collaborating, etc.

Do you have any additional comments?

Thanks!

Thank you for participating in this questionnaire.

Hopefully this will help us compare experiences, learn from each other, identify common pain points, and improve collaboration on MXCuBE.