MXCuBE developers’ meeting
29 April 2021

Participants:

- Marcus Oscarsson, Antonia Beteva, Jean-Baptiste Florial, Daniele de Sanctis (ESRF)
- Meghdad Yazdi,(MAX IV)
- Jordi Andreu (ALBA).
- Lais do Carmo (LNLS)
- Rasmus Fogh (Global Phasing)
- Roberto Borghese (Eletttra)
- Martin Savko (SOLEIL)
- Michael Hellmig (HZB)

Apologies: Ivars Karpics (EMBL Hamburg)

Minutes: Rasmus

Additions to agenda

Next MXCuBE meeting, Release procedure, Queue refactoring,

Site status

The site status overview was skipped at this meeting

Release procedure

JA summarised his Googledoc overview of release procedures. Key points were:

- Agreement on working practices
- Semantic versioning
- A not too complex system
- Version pugrades. Bumpversion (NB this program is deprecated)
- Changelog
- Fit to Github

RF had some reservations. MXCuBE was unusual since the github repository was **only** used by developers, and each site had its own locally adapted version that was sometimes far behind, but still actively developed. The concept of ‘release’ did not have its usual meaning. We agreed on semantic versioning, which limits what can be changed when. The main question was which operations (tests etc.) needed to be done before a version increase, to make it easy to get hold of the versions and stay consistent with them as you made local changes, and to have procedures that made it easy and sure to do the right things. It might be a good idea to have a separate release branch to make it clear what went to release and when. The current master-is-the-only-branch procedure had the complication that there were often part-completed, partially tested commits, and that it could be tricky to ensure synchronisation between mxcube and HardwareRepository repositories.
AB, MS felt it was less important whether we called it a ‘tag’ or a ‘relase’ as long as we had the procedures.
JA pointed out that a tag was only a github thing; we would need to specify and cahgne working practices. Also, when should we bump a versoion – surely not **every** commit? Anyway, it was not a question only of tagging, also of putting the result in Pypi, conda, … How would we handle major refactoring like for the queue?
The question fo site-specific code was raised, and it was generally agreed that the semantic versioning should only refer to the core – e.g. adding a site-specific feature need not cause a bump in the minor version.
There was a discussion on the need for a development branch, so that the move of code from development to release branch would be the time when tests were made and consistency enforced. RF and others were in favour, MO, AB and others against. It was agreed to begin by deciding which steps and procedures had to be done when adding a feature / bumping the version, and deer the discussion of branch structure till later.
MO was volunteered to write a description of the relevant procedures.
JA volunteered to expand the googledocs document on releases.
JA noted that one desired result would be to make MXCuBE and new versions easier to adopt. An important part would be to have a workable mockup version as part of the core.
Lais recommended adding documentation notes, and generally proposed to stick to tags, keep the system simple, and add complications later if required.

Queue refactoring

Since Ivars Karpics was not at the meeting to present the current state of queue refactoring, the point was postponed to a later meeting.
.

Enhancement proposals

Arising from the point on queue refactoring, MO suggested we might consider introducing a more formal procedure for enhancement proposals, to make discussion on similar issues easier in the future.
JA proposed the Sardana Enhancement Proposal template as a good starting point.
It was pointed out that Googledocs were not well suited for multi-participant discussion, and Github was mentioned as an alternative,.
JA noted that using github and enhancement proposals generally was very useful for multisite collaborations, it only beocmes a bit burdensome for in-house work.
SSX might be a good example of where enhancement proposals would be helpful.
DdS noted that the first step of any enhancement would anyway have to be prototyping, which is less collaborative. So when should you take to EPs?
JA suggested to do a prototype first, then write an EP.
The meeting agreed that introducing Enhancement Proposals would be a good idea.

Next Meeting

The date for the next developers’ meeting was left for MO to organise, towards tMid-June.