Meta-release Process

This section explains the processes for meta-release planning, milestone scheduling and progress reporting on the planned a meta-release.

- Meta-release plan
- Meta-release schedule
- Release contacts
- Planning and progress reporting on a meta-release
  - Release Management
    - Meta-release tracking
    - Milestone activities
    - Milestone status
  - Commonalities and ICM
    - Release tracking
    - Milestone activities
    - Version and release numbering
  - API Sub Projects
    - Release tracking
    - Milestone activities
  - TSC

Meta-release plan

For each meta-release, a meta-release plan provides the dates and progress of the meta-release. It covers the status of

- Commonalities and ICM releases
- The release of the API versions that are planned to be included in the meta-release.

The list of meta-release plans (or meta-releases for short) can be found here: CAMARA meta-releases.

Meta-release schedule

CAMARA meta-releases are scheduled twice per year at approximately 6 month intervals (March and September).

Meta-releases are named SpringYY or FallYY respectively, where YY is the (short) year number. For example Fall24, Spring25, Fall25, etc.

A meta-release cycle covers 7 milestones, M0 through M6. The below figures provide the typical planning and milestone dates for a Spring or Fall meta-release.

The “typical” meta-release schedule for Spring and Fall meta-releases are depicted in the below figures. NOTE: Actual dates for a given meta-release shall be managed on the dedicate meta-release plan which are listed here: CAMARA meta-releases.
Release contacts

Meta-releases are administered and tracked by

- the CAMARA Release Managers, supported by
- the release contacts of the Commonalities and ICM working groups
- the release contacts of the API Sub Projects

The API Sub Project teams shall put at least 2 contacts on their API release trackers.

RELEASE CONTACT LISTS

The contacts for the Commonalities, ICM and Release Management Working Groups can be found here: CAMARA meta-releases

The contacts per API can be found on the meta-release plans.

All release contacts need to subscribe to the release management mailing list.

All communication on an ongoing meta-release and its progress will be managed through this mailing list, with in copy the maintainers mailing list (link to be added).

M0 and M5 will be announced on the all@lists.camaraproject.org.

Draft messages for the different announcements are prepared on the following page: Release Management announcements

Planning and progress reporting on a meta-release

The planning and status of the ongoing meta-release is discussed in the Release Management working group meetings.

The meta-release planning and status is visible on the meta-release plan created for each meta-release. The meta-release progress data is obtained as follows:

- the Commonalities and ICM data is updated directly on the meta-release plan by the respective teams.
- the API data is managed by the API Sub Project teams on their API release trackers and pulled into the meta-release plan automatically.

The milestone activities are the activities between the milestones requested from the various teams to plan and provide visibility on the progress of a meta-release. They are described in detail for each team below and are also listed at a higher level in the Meta-release Milestones table.

NOTE: in case of differences between this page and the milestone table, this page takes precedence. In case of such differences or missing points or proposed changes or other feedback, please use the Meta-release feedback page.

Release Management

Meta-release tracking

To be able to track a meta-release, the Release Management team first create the meta-release plan under the CAMARA meta-releases page as follow:

- Go to the following page Meta-release <meta-release name>, and create a copy, changing the parent page to: CAMARA meta-releases (make sure to tick the “copy files and images” box).
- Follow the instructions in red on the copied page.

This results in the new meta-release plan to be used for tracking the meta-release.

Milestone activities

- Request the TSC to declare the meta-release kick-off.
- M0
  - Announce the meta-release kick-off on the CAMARA mailing list (all@lists.camaraproject.org). This announcement indicates:
    - the Commonalities and ICM teams to prepare for M1 the scope definition of what they plan to put in the meta-release
    - all API Sub Projects to create the API release tracker for their next planned API version(s) as described here: API release tracking process.
    - the Outreach Committee to start preparation of marketing activities based on information in the meta-release plan
  - Once available, check the final alpha releases PR of Commonalities & ICM and their release asset availability. If OK, submit to TSC for approval.
  - After TSC approval, announce M1 on release management and maintainers mailing lists.
- M1
  - Check for Commonalities and ICM that all release assets are available in their final release-candidate PRs.
  - If OK, request TSC to approve creation of the final release-candidate of Commonalities and ICM
  - After TSC approval, and update of the meta-release plan with the final release-candidate tags of Commonalities and ICM, announce M2 milestone.
- M2
  - From M1, for all APIs, once available, check the first API release-candidate PR and, if OK, approve PR.
  - Announce M3 milestone once all approved API release-candidates have update their API release tracker.
• M3
  - For all APIs, once available, check final API release-candidate PR and, if OK, approve PR. Approval can start as soon as an API Sub Project indicates "M4: ready for RM" on the API release tracker in the comments field. It is not necessary to wait for the actual M4 date if an API release is ready before.
  - For all APIs, once available, check API public-release PR, and, if OK, ask for formal TSC approval.
  - Announce M4 milestone when all API public-releases are approved by TSC.

• M4
  - Check that all API release trackers are updated with their public-release tag for the meta-release.
  - Propose meta-release content to TSC.
  - After TSC approval, publish the meta-release.
  - Announce the M5 meta-release on all@lists.camaroproject.org

• M5
  - Review the release process with all teams and identify areas for improvement for the next meta release with all teams. A meta-release feedback page is available to all to add comments, feedback and suggestions for improvement.
  - Propose improvements for TSC approval to TSC to be implemented for the next meta-release.
  - After TSC approval, announce M6 milestone

The actual milestone dates and status shall be updated on the meta-release plan when the milestone is achieved.

Milestone status

The overall milestone status is reflected by the milestone status table on the meta-release plan. Its content is managed by the Release Management team and shall have values defined as follows:

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Before milestone</th>
<th>After milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>M0</td>
<td>kick-off preparation</td>
<td>kick-off done</td>
</tr>
<tr>
<td>M1</td>
<td>alpha PR</td>
<td>alpha for M2</td>
</tr>
<tr>
<td>M2</td>
<td>release-candidate PR</td>
<td>release-candidate for M4</td>
</tr>
<tr>
<td>M3</td>
<td>API first release-candidate PR</td>
<td>API first release-candidate for M4</td>
</tr>
<tr>
<td>M4</td>
<td>API final release-candidate PR</td>
<td>API final release-candidate</td>
</tr>
<tr>
<td></td>
<td>API public-release PR</td>
<td>API public-release for M5</td>
</tr>
<tr>
<td>M5</td>
<td>meta-release preparation</td>
<td>meta-release published</td>
</tr>
<tr>
<td>M6</td>
<td>retrospective preparation</td>
<td>retrospective done</td>
</tr>
</tbody>
</table>

Commonalities and ICM

The Commonalities and ICM teams shall respectively update and maintain their information directly on the active meta-release plan accessible here: CAMA RA meta-releases (select the active meta-release from the list).

Release tracking

The release tracking for Commonalities and ICM is done by updating the Commonalities and ICM table available on the meta-release plan of the active meta-release.

The following explains the fields of this table and when / how it needs to be updated by the Commonalities and ICM teams:

<table>
<thead>
<tr>
<th>Field</th>
<th>Field explanation</th>
<th>When to update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target version</td>
<td>the public-release version targeted for the meta-release</td>
<td>at M0 or when the target scope issue is created</td>
</tr>
<tr>
<td>Target scope</td>
<td>link to GitHub issue defining the release scope</td>
<td>when the issue is created</td>
</tr>
<tr>
<td>M1 date</td>
<td>actual final alpha release milestone date</td>
<td>when alpha release is created (before M1)</td>
</tr>
<tr>
<td>M2 date</td>
<td>actual final release-candidate milestone date</td>
<td>when public-release is created (before M2)</td>
</tr>
<tr>
<td>(Pre)release tag</td>
<td>link to latest pre-release</td>
<td>when a new pre-release is created</td>
</tr>
<tr>
<td>M5 date</td>
<td>actual final public-release milestone date</td>
<td>when public-release is created</td>
</tr>
<tr>
<td>Public-release tag</td>
<td>link to final public-release</td>
<td>when public-release is created</td>
</tr>
</tbody>
</table>

Milestone activities

The activities of the Commonalities and ICM teams during the release cycle are the following:
• **before M0:**
  - Starting from the M2 of the previous meta-release, prepare the scope definition for the upcoming meta-release.
  - Start implementing the scope through one or more alpha releases.

• **M0**
  - As soon as possible after M0, fix the scope definition for the release for the meta-release:
    - Record scope in a dedicated GitHub issue, e.g. called "Fall24: Commonalities / ICM scope".
    - Submit scope issue for TSC review
  - Develop final Commonalities & ICM scope through one or more alpha releases
  - Update data in the meta-release plan with each alpha release
  - Create final alpha release PRs and submit to Release Management
  - After TSC approval, create approved final alpha release for Commonalities & ICM
  - Update the meta-release plan with the alpha release tag

• **M1**
  - Fix bugs raised by users through one or more release-candidates
  - Update data in the meta-release plan with each release-candidate
  - Create final release-candidate PR and submit to Release Management for checking
  - After TSC approval, create approved final release-candidate for Commonalities & ICM
  - Update the meta-release plan with the final release-candidate release tag. The final release-candidate shall be used by the API Sub Projects to work with for their API release-candidate development.
  - Start the scope definition and alpha release for the next meta-release.

• **M3 and M4**
  - During API development, changes to the final Commonalities or ICM release-candidates may be requested, leading to new release-candidates to be reflected on the meta-release page.
  - The public-release PR of the Commonalities and ICM will only be created latest 2 weeks before the M4 date. Once committed, these will also be used for M5 for inclusion in the meta-release.
  - Fix bugs raised by API testers through one or more release-candidates
  - Create public-release PR for Commonalities & ICM
  - After Release Management check and TSC approval of the public-release PR
    - Create the public-release
    - Update the meta-release page with public-release tag

• **M5**
  - Provide feedback on meta-release

• **M6**

**Version and release numbering**

**PROPOSAL FOR DISCUSSION**

**Version numbering**

For Commonalities and ICM, the version is defined here ????

*Open question: is there any place where the version is captured ? a config file or other ?*

The version numbering scheme follows Semantic Versioning 2.0.0.

**Release numbering**

A Commonalities or ICM release is tagged with a GitHub release tag named as follows:

- it is numbered with the version number: x.y.z
- is prefixed with "r" in the release tag: rx.y.z
- for alpha or release candidates it is post-fixed with the version extensions: rx.y.z-alpha.m or rx.y.z-rc.n
- for public-release, no version extension is allowed

The following table provides the overview of the version releasing and numbering scheme through the release cycle.

<table>
<thead>
<tr>
<th>Stage in release process</th>
<th>Version number</th>
<th>Release number / release tag</th>
<th>Version can be released</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha</td>
<td>x.y.z-alpha.m</td>
<td>rx.y.z-alpha.m</td>
<td>Yes (internal - M1)</td>
</tr>
<tr>
<td>release-candidate</td>
<td>x.y.z-rc.n</td>
<td>rx.y.z-rc.n</td>
<td>Yes (internal - M2)</td>
</tr>
<tr>
<td>public-release</td>
<td>x.y.z</td>
<td>rx.y.z</td>
<td>Yes (M5)</td>
</tr>
</tbody>
</table>

**API Sub Projects**

**Release tracking**

API Sub Project teams shall create and update the API release tracker for each of their API version(s) as follows:

- Create the API release tracker as described here: API release tracking process
- With each alpha or release-candidate API version, the API version and release tag shall be updated.
- The actual milestone dates shall be put in the release tracker when the milestone is passed.
- When M4 is successfully passed, the link to the public-release tag shall be added.

An API version of an API Sub Project should show on the meta-release page as soon as an API release tracker has been created under the API Sub Project's API release tracking page for the API version planned to be released.
If an API release is not visible, please check that the correct meta-release label is added to the API release tracker page.

**Milestone activities**

The following are the actions for milestones:

- **M0**
  - Start following the scope definitions of Commonalities and ICM and assess impact on API version
- **M1**
  - Create API release tracking page for the API if it does not yet exist
  - Create API release tracker for the API version to be released
  - Define scope of API release:
    - Record scope in dedicated GitHub issue.
    - Update the release tracker with the scope issue link
  - Develop API scope through one or more alpha releases
  - Update the API release tracker with each alpha release
  - Create first release-candidate PR and submit to Release Management
  - After Release Management approval:
    - Create first release-candidate for the API
    - Update the API release tracker with the release-candidate tag
- **M3**
  - Fix bugs raised by API testers through one or more release-candidates
  - Update API release tracker with each release-candidate
  - Submit final release-candidate PR to Release Management for checking
  - After final release-candidate PR approval by Release Management:
    - Create final release-candidate and update API release tracker
    - Create API public-release PR
  - After TSC approval of the PR:
    - Create public-release
    - Update the API release tracker with public-release tag
- **M4**
  - N/A
- **M5**
  - Provide feedback on meta-release
- **M6**

**TSC**

- **M0**: Declare meta-release kickoff
  - Approve Commonalities and ICM scope
  - Approve final alpha release PRs of Commonalities & ICM
- **M1**
  - Approve final release-candidate PRs of Commonalities & ICM
- **M2**
  - Review scope of selected APIs (case by case selection)
- **M3**
  - Formal approval of the Commonalities & ICM public-release PRs
  - Formal approval API public-release PRs
- **M4**
  - Meta-release approval
- **M5**
  - Provide feedback on meta-release
  - Meta-release improvements approval
- **M6**