OPNFV Release Process 2.0 (DRAFT)

THE LINUX FOUNDATION

LF NETWORKING

Considerations

- Reconcile CNTT requirements with OPNFV
- Simplify and reduce the number of milestones
- Support OPNFV level requirements planning
- Improve release planning at the project level
- Improve accountability across all project types
- Enable project self releases independent of OPNFV cadence
- Increase community engagement in the release process

Assumptions

- Per the mission statement, OPNFV has objectives that require a coordinated release effort, as directed by the TSC, and is not merely a collection of siloed projects.
- The OPNFV TSC supports a regular OPNFV release
- Supporting CNTT requirements will be the highest priority for OPNFV for the foreseeable future
- Projects may veto requirements that affect their project and engage in negotiations with requirements stakeholders over blocking issues, such as resource constraints.
- The TSC wishes to continue to include projects that are consistent with OPNFV objectives but are not dependencies of CNTT related work product.
- This release process is a starting point and is expected to evolve over time as details are added and as the community gains experience implementing it.

Overview

- This release process proposal is for a “meta-release,” or a “release of releases”.
- It establishes a requirements process that enables OPNFV to meet stakeholders’ requirements that may span multiple projects.
- However, it also enables individual projects or groups of projects to follow their own release model (e.g., continuous delivery) and to self-release at their own cadence, via a self-release process.
- Projects which are dependencies of CNTT work product will follow a gating process, overseen by an integration project.

OPNFV Level Requirements Planning

- Requirements planning is a critical part of this release process proposal and represents the biggest change from the previous process
- Why is requirements planning important?
  - We need a way to address CNTT requirements affecting multiple projects.
  - We need a way to agree upon and to prioritize broad requirements that help to advance our mission, or to respond to a concern that affects most projects.
    - Example: Python 3 migration

OPNFV Level Requirements Planning

- Overview
  - Requirements working group or subcommittee gathers and vets requirements from multiple stakeholders and makes recommendation to the TSC
  - OPNFV Release Requirements are reviewed and approved by TSC at each milestone
    - Requirement is de-scoped if not approved by the TSC
  - Projects agree to prioritize OPNFV level requirements over other work
  - Each requirement has an owner and is documented in JIRA
  - TSC approval at M1 requires support commitment for each requirement from relevant projects
    - Project support is documented in project release plan

What does “vetting” mean

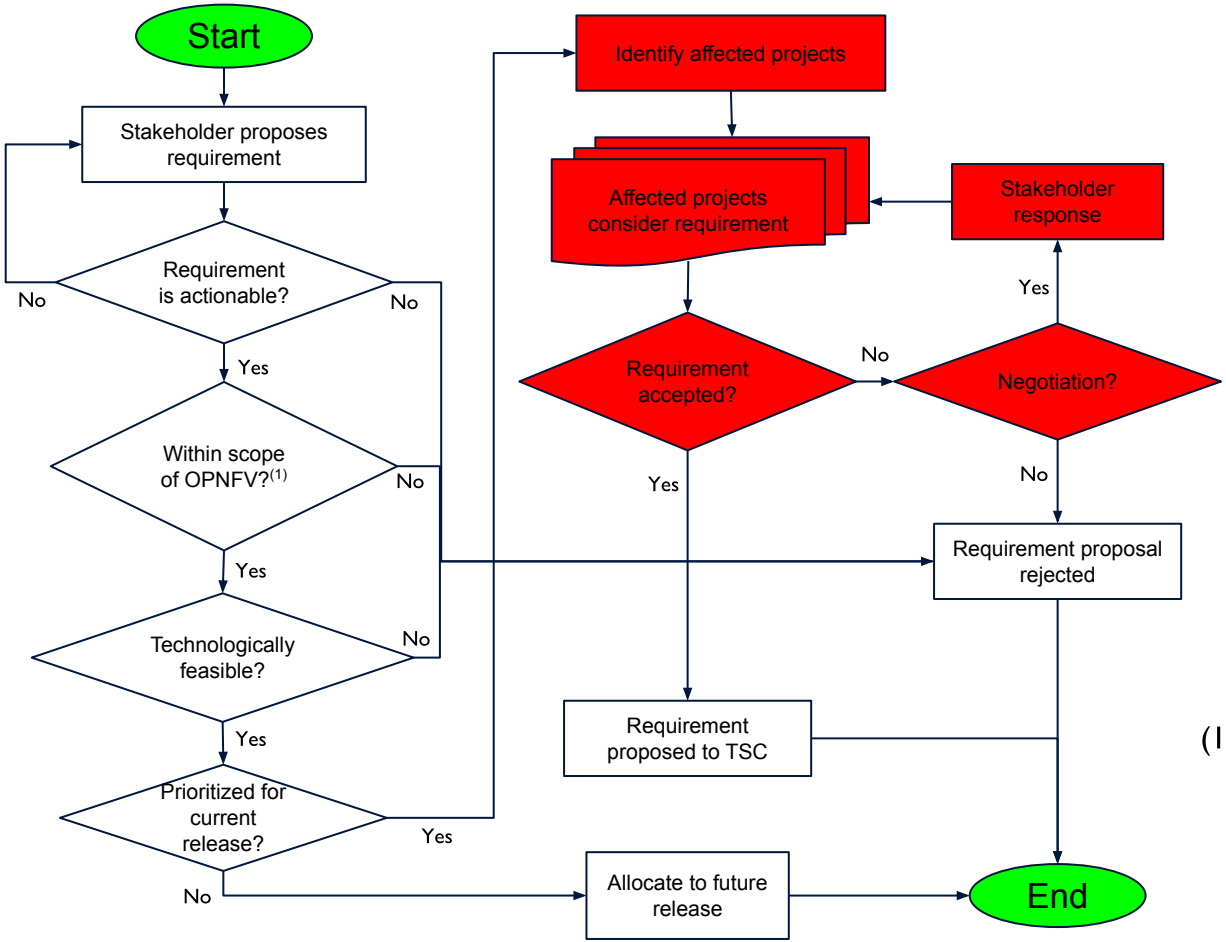
- A requirements group is needed to vet proposed requirements. Why?
  - We want to focus our time and energy in a given release cycle on requirements that:
    - Are actionable, i.e., clearly defined with sufficient detail for implementation
    - Are within scope, and meet OPNFV objectives, as defined by the Charter, the Governing Board, and the TSC
    - Are technologically feasible
    - *Have support and sufficient resources in the affected projects as agreed to by the PTL*
    - Have priority relative to other eligible requirements
- The requirements group will apply the above criteria to requirements proposed by stakeholders and make a final proposal to the TSC.

More on verifying support among affected projects

- The requirements group helps stakeholders identify and contact the affected projects
- The affected projects consider the proposed requirement and indicate whether they will support the requirement, or not
- If the project does not support the requirement, for example, due to a shortage of resources, then this may start a negotiation between the stakeholder and the project.
- If the project does not support the requirement proposal, then the requirement will not become part of the release requirements.
- Note that the requirements group *facilitates* communication with affected projects. It's up to the stakeholder to determine support.

Requirements Subcommittee

- Review requirement proposals submitted by stakeholders
- Verify that requirements have support from affected projects
- Verify that requirements have an owner and are well documented in JIRA
- Allocate requirements to current or future releases
- Recommend a set of prioritized requirements to the TSC for approval for the current release



(1) Mission Statement

Requirements Owner Responsibilities

- The requirements owner is an active participant throughout the release.
- The requirements owner will:
  - Prepare and present a detailed, well defined requirement proposal for consideration by the requirements subcommittee (documented in JIRA).
  - Respond to questions and direction from the requirements subcommittee
  - Work with the requirements subcommittee to determine which OPNFV projects are affected by the requirement
  - Engage with the affected projects to determine whether they will support the requirements. If possible, negotiate issues such as resource constraints.
  - Once the requirement is accepted by the TSC for the release, monitor and report status back to the TSC at each milestone, or upon request.
  - Determine a test plan and test lead, as necessary.
  - Work with the projects to overcome blocking issues in order to successfully complete the requirement for the release.

Requirements for All Projects

- OPNFV repo
- Project release plan conforming to OPNFV template
- CI using OPNFV resources
- Documentation and release notes complying with OPNFV documentation guidelines
- Release testing
  - Projects that are dependencies of CNTT will be required to pass integration testing (e.g., Functest, Airship, CIRV), which will be overseen by an integration project or manager
  - Projects that are not dependencies of CNTT may self-verify

Project Release Plans

- Template based
- Reviewed and approved as part of release process
- Commitment to OPNFV-level requirements
  - Document how requirement will be met
- Other project objectives for the release
- Specify deliverables
- All work documented in JIRA and assigned to release

Self Release

- All projects will maintain internal versioning
  - OPNFV release versioning will follow current practice of using the prefix “opnfv-” on version numbers to distinguish them.
- Projects will be encouraged to contribute to OPNFV releases, approximately every 6 months, that meet OPNFV requirements established by the TSC and the release process.
- In addition, projects may release independently, using a common Self-Release Process
- The self-release process will enable community access to self-release work product in a common way

Documentation

- The current documentation is organized around the traditional OPNFV concept of “scenarios,” which is no longer a prominent aspect of OPNFV.
- Need to reconcile CNTT documentation with OPNFV documentation organization and process.
- Ask the DOCS project to lead an effort, along with other stakeholders, to develop and propose new documentation structure to the TSC.
- Continue current practice of having milestone requirements for preliminary and final documentation as part of release process.

Integration Testing and Gating

- Projects that are dependencies of CNTT will be subject to integration testing and gating.
- An integration project or manager will track project test and integration status, and will report this information to the release manager and to the TSC.
- Release Candidate milestones will be based on gating status and approval by the integration project or manager.

Milestones

- Projects must complete tasks at each milestone to be approved to proceed in the release
- Requirements are evaluated at each milestone to determine whether they remain feasible
- Milestones:
  - M0 - Start of Release
  - M1 - Planning Complete
  - M2 - Readiness Review
  - RC0 - First Release Candidate
  - RCn - Final Release Candidate

Milestones: Planning

- Requirements submitted, reviewed, and approved by TSC
- Project release plans completed, reviewed, and approved
- All work planned for the release is documented in JIRA and assigned to the release (fix version field)

Milestones: Readiness Review

- Resolve high priority JIRA issues
- Complete preliminary documentation
- Marketing
  - Complete project update message on designated wiki page:
    - 1. What has changed
    - 2. Why significant/beneficial to a user/consumer of OPNFV?
  - A release marketing messaging document will be derived from these project updates and presented to TSC for review/input

Milestones: Release Candidate 0, 1, …, x

- Approval by integration manager that integration requirements have been met
- Verify release plan, including all planned testing, has been completed
- Prepare and verify release artifacts
- Complete final documentation
- Complete tagging
- Marketing feedback integrated and presented to the TSC

ToDo - Prerequisites for initiating new process

- Establish requirements working group
- Create requirements JIRA template
- Establish integration management project
- Determine new documentation layout/organization
- Develop self-release process
- Develop project release plan template
- Develop and implement integration gating

Frequently Asked Questions (FAQ)

Q: What do you mean by “requirements owner” or “stakeholder”

A: These would typically be organizations or groups who use OPNFV software, or are a potential user (e.g., CNTT). It could also be any organization who has an interest in standards compliance or security issues. Requirements owners could be either internal to OPNFV or external. An example of an internal requirements owner could be the Closed Loop Automation working group, which might have a use case that they would like to implement that requires contributions from one or more OPNFV projects. A requirement could also come out of the community, such as migrating to Python 3.

It’s important to keep in mind that the requirements owner must be engaged for the entire release (see Requirements Owner Responsibilities). The requirements owner may also need to provide developer or test resources if they want the requirement to be accepted by the relevant projects.

Examples: CNTT, TAC, EUAG, CNCF, GSMA, ETSI, ONAP

Frequently Asked Questions (FAQ)

Q: Does this process impose a lot of new requirements on projects and PTLs?

A: This process asks that projects do the following new things:

1. Complete a project release plan based on a template, rather than free-form
2. Review stakeholder requirement proposals if they affect the project
3. Agree to prioritize OPNFV requirements (previously approved by the project) ahead of project plans.

Example: CNTT submits a requirement that requires work by Functest. The requirement is agreed to by Functest during the requirements phase, approved by the requirements subcommittee, and approved by the TSC. Meanwhile, Functest also has plans to do some refactoring as part of the release. In this case, Functest agrees to prioritize the CNTT requirement ahead of their plans for refactoring.

Frequently Asked Questions (FAQ)

Q: Does this process require projects to follow a “waterfall” release model?

A: No, as mentioned at the start of the presentation, this release proposal is for a meta-release, or release of releases. The process does not require individual projects to follow a particular release model. Projects, or groups of projects, are free to follow whatever model they choose. In addition, outside of the OPNFV releases, projects are free to set their own cadence and use a common self-release process.

Frequently Asked Questions (FAQ)

Q: Does the TSC impose stakeholder requirements on the projects?

A: No. As described earlier in the presentation, projects have an integral role in determining requirements for the release. If a stakeholder requirement affects a given project, that project MUST agree to the requirement before it becomes a release requirement. This may involve negotiation between the project and the stakeholder over developer resources, for example.

BACKUP

OPNFV Level Requirements Planning (M1)