

20191203 [CNTT - RLM] - Current State Discovery - Meeting Minutes

[Issues with OPNFV Release Process.pdf](#)

The following notes were captured as part of the CNTT Release & Lifecycle Management gap analysis initiative, current state discovery meeting.

What are the OPNFV projects?

- Projects who have joined OPNFV
- Project could be anything related to NFV

OPNFV Organization

- Sub-organizations/characteristics
 1. Release Management
 2. Projects
 - a. Types
 - i. Feature
 - ii. Installer
 - iii. Test Frameworks
 - b. Characteristics
 - i. Interdependent
 - ii. Stand-alone
 - c. Other?
 3. TSC
 - a. Approves process schedule and exceptions
 4. Release Engineering
 - a. Has dual-hat of being on LF Infrastructure team
 - b. Manages and supports host infra
 - c. Trevor Bramwell / Aric Gardner - LFN employees assigned to OPNFV project)

Release Management Processes (good example is Hunter: <https://wiki.opnfv.org/display/SWREL/Hunter>)

- Primary organizing units: Scenarios
- Scenario Development
 - Feature projects are dependent on installer, installers are dependent on upstream component (OS, K8s, SDN controller) interrelationships
 - Stand-alone projects do their own thing but are not dependent on other projects
- CI/CD
- Technically OPNFV doesn't have a code freeze, but does have a feature freeze at milestone 5
- Post-Release
- OPNFV end-user (nobody really knows who that is. Clearly people are doing downloads, but not sure who.)
- AI - Talk with Heather and Brandon about how folks are using OPNFV.
 - Large chunk of what it gets used for is CI.
- Tool chain probably provides the most value (installers, FuncTest, Yardstick, etc.)
 - NFV Bench, Doctor, Barometer get significant attention as stand-alone projects also

Technologies

- Test Case Database (Cedric knows technology)
 - Cases are associated with tools needed for the project they are related to.
- Jenkins
 - Scenarios are used to prove out installers (e.g. can deploy past OS health check?)

Highlights

- Upstream / Installer related
 - Significant amount of time and focus on adapting installers to latest version of OpenStack. OPNFV always seeks to integrate with latest release of OS (so OPNFV can provide that feedback to them and make corrections), but using latest release is source of installer issues – David is working on presentation of this issue to TSC.
 - Complaints:
 - Quality / necessity of artifacts
 - Release too late to provide useful feedback to cutting edge development on upstream projects (e.g., 8+ months after OS release)
 - Producing installers was not original intent of OPNFV
 - Installer focused method of doing release management
 - 30-40% of release cycle efforts
 - Two objectives are: developing reference implementations vs. providing timely feedback to upstream communities.
 - Need to be careful: for example--today's discussion around version of Airship using Ocata. Ocata (version?) is old.
 - *David McBride - yes, this is not the usual approach for installers participating in OPNFV. Typically, the installers adapt to the latest released version of OS.*
- The release process is focused on scenarios and installers, not stand alone projects (but some stand-alone projects are adding significant value but not getting visibility for it)
 - *David McBride - that's correct. Not only lacking visibility, but also accountability and transparency, since the release process is not geared to track them in any detailed way.*
- Projects having difficult time getting deployed and having reliable results (Milestone 3 very important)

- Installer platforms unstable (installer depends on OS integration)
 - Installers were a means to an end, not original intent of OPNFV focus
- Conflict of objectives: provide reference implementation but also provide rapid feedback to OS
- Wasn't uncommon for OS to be unstable for a month or two before engineers able to work with it (40% of effort)
- **Release dates adjusted based on ability to meet stable date**
 - Multiple installers participating all trying to get stable together
 - Needed so that feature projects could reliably do performance and testing
- Test frameworks like FuncTest need to be compliant with Installers (Milestone 3 also)
- OPNFV – at one-time XCI operating as optional gate for OpenStack commits due to good reputation with upstream communities. OPNFV based-tests triggered by OS commits in XCI environments. So OPNFV was providing great value to OS.
 - Ildiko OS rep to TSC.
- Issue – gap between stable branch and end of release.
- Confusion by having a second release guaranteed after release date (JB – maybe need to call it 'stabilization period' but no major releases—only 'fixes' after major release date)
- Accelerator-- if we as a community could agree on a neutral means of deploying a stack (neutral from market or vendor stand point) – how was able to ONAP able to do this?
- Not doing a good job of project level release planning.
 - Projects not required to clearly define what they are doing. Objectives. Deliverables.
- No formal way to drive OPNFV-wide requirements to projects.

What ifs

- What if OPNFV was strictly focused on CI/CD? Only 'be' the feedback loop.
- What if OPNFV was strictly focused on stable reference implementations? Use only stable releases.

Because ,OPNFV tried to do both unable to satisfy both. ([David McBride - yep, key takeaway, IMHO](#))

Action Items

- Talk with release engineering (releng)
- Talk with Cedric re: tools and technologies used in test case management