Issues with OPNFV Release Process

THE LINUX FOUNDATION

Conflict between objectives

- OPNFV has two objectives:
  - Develop reference implementations
  - Provide timely feedback to upstream communities
- These objectives are in conflict
  - Stable or latest?
    - Reference implementation depends on stable releases from upstream
    - Timely feedback implies working on latest code
  - Release artifacts?
    - Reference implementation implies stable release artifacts consumable by end-user
    - Timely feedback implies patches and JIRA tickets for upstream components and no need for release artifacts.

Installer focused

- Three years ago, most projects were part of a “scenario”:
  - Installer
  - Feature integrated with installer
  - Test framework integrated with installer
- Today, only a small number of projects are part of a scenario, but release process is still focused on this configuration
- Low visibility of standalone projects not part of a scenario
- 30% - 40% of release cycle dedicated to:
  - Installer integration with OpenStack
  - Installer stabilization

Project level release planning is undefined

- No requirements on project level release plans
- This means that objectives and deliverables are unclear
- Very little accountability for projects, since objectives and deliverables are undefined, especially if they are a standalone project not tied to a scenario
- No formal mechanism to drive common, OPNFV-wide requirements to projects, e.g.,
  - Use of common upstream components
  - Python2 ⇒ Python3 transition
  - Container usage