OPNFV Simultaneous Release 1.0 Planning

Discussion Notes
Dave Lenrow

Sample V1.0 Milestones

Milestone	Offset 0	Offset 1	Offset 2	Events
M0	November 1, 2014			Simultaneous Release Open
M1	December 1, 2014			1. Projects Must Have Declared their intent to participate in the simultaneous release 2. Participating projects must have published a candidate Release Plan for public comment (Release Plan Template)
M2	December 14, 2014			Participating Projects must have declared their final Release Plan
M3	Jan 10, 2015			Latest possible Continuous Integration Test Start
M4	April 1, 2015			1. API Freeze 2. Latest possible Continuous System Test Start
M 5	April 15, 2015			1. Code Freeze (bug fixes only from here) 2. String Freeze (all internationalizable strings frozen to allow translation time) 3. Latest possible date for commencing User facing Documentations
RC0	May 8, 2015			
RC1	May 15, 2015			
RC2	May 24, 2015			Participating Projects must hold their Release Reviews, including User Facing Documentation.
Formal Release	May 31, 2015			

How is OPNFV SR 1.0 unique?

- Majority of release artifacts built on sources that live upstream
- Need to demonstrate particular performance or scalability achievements?
- Goal to jumpstart ecosystem of VNF developers
- Assumes infrastructure and CI process that does not yet exist.
- Need to include at least one useful demo/POC composing and creating service chain of VNFs/

How to join the SR

- Ask whether proposed project is part of platform or built above
- Propose that SR is platform only (other releases can be supported too)
- Elect a project lead who is primary contact in community for issues
- Declare intent by milestone date, define dependencies and project external interfaces
- Provide SR plan by milestone date
- Get SR plan accepted/approved
  - Who/How
- Hit milestones to remain in compliance with SR requirements
  - Start of CI milestone wth adequate test coverage is critical

Some SR Requirements to consider

- Documentation – How to measure compliance? Metrics? Ratings?
- Interface
- Test Coverage: Intra-project? Inter-project? Sonar or tool based metrics? Are there tools across all components parts (java, C, C++, python, bash, etc?)
- How does test requirements project fit in? Other infra projects?
  - LF code and tooling vs OPNFV code and tooling?
- Project must commit to supporting N x stable/maintenance releases

First SR assumptions

- Keep it simple. Deliver basic platform and tools
- CI will not include full automated running of all necessary test coverage so there will need to be a separate qualification process in addition to CI
- Significant time should be allowed for new/closed bug count ratio to converge

What to Include

- N x Upstream
- VF optimization projects?
- NFV specific scripting/automation
- Exclude
  - All but test infrastructure VNFs
  - Leaf projects in general (No benefit to platform consumers of integrated leaves, unnecessarily constrains leaves release/deploy)
  - Infra projects too immature to have dependents (policy deploy,

SR mechanics

- Multiple offsets to allow quality convergence by layers?

Latest ODL Lithium SR plan

- https://wiki.opendaylight.org/view/Simultaneous_Release:DRAFT_Lithium_Release_Plan_ckd

Questions bigger than just SR

- How do we carry changes to upstream projects that are newer than their stable/released branches?
- Are we committed to a continuous delivery model, rather than just CI?
  - Is there some aspect of test that will never be part of the per-patch, automated CI regression process?
  - Minimal per-SR work that isn’t business as usual for devs, and test resources?

Preliminary list of included projects

- Upstream

- OPNFV

- Patches