ONAP
OPEN NETWORK AUTOMATION PLATFORM

ONAP Beijing Lessons Learned

Partners in crime

@Catherine Lefevre, @Gildas Lanilis, @Seshu Kumar M, @Eric Debeau, @Alla Goldner, @Helen Chen,

@Chris Donley, @Stephen Terrill, @Kenny Paul, @Jason Hunt, @Brian Freeman, @Tim O'Keefe Alex Vul, @ramki krishnan

Wednesday, June 20, 2018

Disclaimers

- What is presented in this deck is what was discussed during the 2 Lessons Learned sessions
  - for the sake of time
  - keep momentum
  - More Lessons Learned
- 60+ people attended the Lessons Learned sessions on Monday and Tuesday

Items Covered

- Communications
- Labs and Test Environment
- Release Management
- Jira
- Code Review
- Development Infrastructure

Communication – Wiki, Mailing Lists, IRC (1/2)

- RocketChat:
  - Need absolutely clear and unambiguous validation that **RocketChat** can be used via https from China and most companies without requiring a VPN, or the use of an alternative device/network. -Kenny: Must be secured if setup by LF (simply b/c of well known pwd).
- WIKI Help:

  things are hard to find unless you type the right keywords or have the proper URL.

  - Some information no more accurate
  - Matter of **organization**, not matter of quantity of information

Communication – Wiki, Mailing Lists, IRC (2/2)

- Sub-Committees → TSC
  1. Is there a good enough flow of information-communication between sub-committees and projects? Overall: yes.
  2. Sub-committee feedback toward the TSC is not systematic (as we wish).
- Sub Committees
  1. Expectation from sub-sub-committee back to the team and then driving the execution
  2. Use-case owner is missing
  3. Use-case team to fill a checklist? We may defined what that role is.

Labs & Test Environment

1. Despite the effort made by some ONAP partners in providing Labs, we have reached the limit on current labs infrastructure.
2. After M4 code freeze, the community is spending a lot of time to get the Health Check sanity test to pass.
3. Need to develop a full Agile CI-CD pipeline. Full Chain to run automatically the sanity HC, CSIT, E2E. Everything running continuously.
4. How to make better usage of XCI-CD environment (OPNFV,...)
5. Supporting HEAT and OOM based deployments is getting harder with duplicate maintenance of the config values
6. Improve CSIT coverage to cover all features that the project delivers and to reduce manual testing (if any)

Release Management

1. Grouping of Projects (staggered - offset): to address Release dependency. Idea of 3 months project (push back at TSC)
2. Defining another milestone (no that good idea?)
3. Start early pair-wise testing?

Jira

1. PTLs are owner of the scope of iteration (pay attention when creating a task and do not add into current iteration backlog)
2. Review the 2 JIRA workflows and define 1 workflow for ALL

Code Review

1. Requires active committers and more active code reviewers (right now only few committers or reviewers are always reviewing and merging the code)
2. Commit process: code review. Dialog necessary.
3. Concerns on big code merge that comes late.
4. In case big code should come in, define an upper timeframe limit.

Development Infrastructure

1. The LF toolchain that is currently in place, do allow to merge in master code that has not been thoroughly tested. This leads to massive disruption in the testing.
2. Nexus 3 slowness. This has been impacting Integration Team tremendously as it tooks 3-4 more times just to download Docker images.
3. Slowness of the full IT chain (jenkins, nexus, wiki,..)
4. Local testing: how easy to setup an env for a developer to perform its own testing before submitting code. Need reference VNF, AAI, ... (too much time spend to setup environment). Idea of lab reference to be used as a model for configuration. (currently SB-07 serves that purpose).
5. Pair wise testing for a great value added in Beijing release.
6. Backup and restore capacity for SB 04-07 in Windriver? Have we ever asked Windriver?
7. Feature parity on LABS (do not over taxe Windriver).
8. Nexus IQ scan performed during the verify. If error then block the build.
9. Idea of parallel on a single job. Currently atomic at build level. To investigate-optimize Jenkins Jobs. (time to build the VMs, ...).

ONAP

OPEN NETWORK AUTOMATION PLATFORM

谢谢