2021-02-04 - Anuket: Release Process Goals and Objectives

Topic Leader(s)

- David McBride
- Al Morton
- Walter Kozlowski
- Scott Steinbrueck

Topic Overview

Brainstorming session on release process goals and objectives. The intent is to get consensus in the community on the high level objectives for a release process, including approval by the TSC, then use those objectives to guide development of the Anuket release process.

Slides & Recording

Minutes

Slides are just a framework to spark discussions. The follow up to this session will be taken into the Anuket TSC. Anuket has a challenge because the artifacts are not only code. They are documents, implementations, etc. Have to explain the concept of a feature from its lifecycle that folds in requirements to conformance platforms end to end. Something that is integrated is important to make it useful to the community. Anuket happened because the components need to be integrated, however it requires changes on how the community thinks about the artifacts and how they can be consumed by the operators and vendors alike. There are artifacts in RM and RA that do not always translate into components that go into RI and RC artifacts. Sometimes the components are forward thinking, can drive use cases and gaps in the technology, that spawns work with peer, downstream and upstream communities – e.g. CNCF Working Group, Barometer project, are influenced by the Anuket use cases and models.

Pierre Lynchhow does the release cycle handle the fact that the downstream artifacts are dependent on the upstream work – RI and RC is dependent on the RA and RM documents. The release cycles need to be staggered to accommodate this. Pierre: RC must be verified vs other CNTT compliant (regarding their testing scope) deployment, else it's chicken & egg (we know that very well in OPNFV) See Functest Gates vsperf gates (under construction?) That's how we work it at ETSI NFV TST. We have (an equivalent to) RA, then once that is ready (OpenAPIs), then we (TST) can develop the tests (RC equivalent). ETSI doesn't really have RI :)
Once RA v 3.3.1 (for example) is ready, then we develop TST (RC) v 3.3.1, released with the same number, but a few months later.

Cedric Ollivier RI is verified by RC, Tom Kivlin as Cedric says, it is the other way round - RI relies on RC. RI would be your develop gates (possibly only mocking in case of ETSI NFV TST). For my understanding the big tent is not that successful tempest is rolling. K8s is very challenging because they use a rolling release schedule, while OpenStack follows a 6 month cycle.

Pankaj Goyalquestion: Does RI "rely" on RC other than for "conformance" (verify that it conforms to the RA)?
David McBrideis proposing a loosely coupled process to support the staggered output.

Pankaj Goyalnotes that the RM and RA artifacts are still valuable to the industry, even if they are not translated into software right away. FYI: While RA's typically follow RM, there can be exceptions. RM is just starting to define Observability, LCM, etc. but RA1 has some needed elements already.

Could potentially skip some releases for the workstreams that are moving at different paces. Heather Kirkseynotes that the stations along the way are the touch points between the workstreams, even if they are moving at different speeds. Scot Steelesays that we should still keep a cadence (6 month), even if the outputs are different and the amount of work done during a given release. Rihab Banday We have run into situations in the past where RI2 has been based on older version of k8s, i.e conforming to older versions of RA2 & RC2. It could be difficult for RI software to keep up with new releases of RA/RC artifacts.

Ildiko Vancsa Release management guide in OpenStack: https://docs.openstack.org/project-team-guide/release-management.html. OpenStack release models: https://docs.openstack.org/contributors/common/releases.html There are different release models within OpenStack to ensure the most efficient work and integration (need to release and maintain libraries, CLIs, APIs, etc). Gergely Csatari Just a bit of clarification to what we just discussed with Beth. OpenStack has a binary project status, the project is either an official OpenStack project or not (and it is decided by the OpenStack TC), however in OIF they have different states, like Confirmed and Pilot, but here it is not clear for me who decides on the state of the projects (my guess is the OIF board).

Ildiko Vancsa Yes, for top level projects (OpenStack, Airship, Kata Containers, etc) it is the OpenInfra Foundation Board of Directors who make the decision when a project gets accepted as confirmed project. From the stage of being pilot for some time first. And as @Gergely said for OpenStack it is the TC (Technical Committee) who can accept a new project team to become part of OpenStack.

Jim Baker : One perspective I'd like to introduce is the role of the Anuket members in the two-phase release process. I think it essential that the spec writers are ACTIVELY involved in the acceptance/approval of the test release. The model of "write a spec, throw it over the wall" does not leverage the talent on the Anuket team. Can we figure out a process that keeps BOTH spec and test writers engaged in each others work product?

Pankaj Goyal : Bullet two needs changes RI doesn't follow RC. We can create an RI based on RA but wouldn't know how well it conforms to the specs -- that is what RC achieves.

Cedric OllivierRC is a test case list + a playbook and somehow a traceability (which is also the test entries). It lists containers and test case names. All the software is included in the containers. RC selects a common test execution framework which helps storing the results.
Al Morton I was writing: , RC is a test case list, the SW implementation of the test cases, and a framework to run the tests, collect results, store the results for later access (public). Pankaj Goyal 1 RC but multiple RI's are possible

Scott Steinbrueckis proposing a release cycle that maps to twice a year, by creating a release package. Each project (workstream) aligns with the package as appropriate for their pace. Discussion on how to fold all the project into the overall cycle properly.

Summary: There is a relatively tightly coupling between RM and RA. The others (RI and RC) are more loosely coupled. Look at OpenStack's experience in managing multiple work steams and projects in a big tent. We should make a decision to move forward. Let's try something and see how it goes. Next iteration will let us improve the process. Use agile methodologies!

Chat Text

06:03:55 From acm : https://wiki.lfnetworking.org/display/LN/2021-02-04+-+Anuket%3A+Release+Process+Goals+and+Objectives
06:04:03 From acm : for minutes
06:17:51 From Cédric Ollivier : RC depends on RA only
06:18:01 From Tom Kivlin : RC does not rely on RI
06:18:14 From Cédric Ollivier : RI is verified by RC
06:18:35 From Tom Kivlin : ^ - as Cedric says, it is the other way round - RI relies on RC
06:18:57 From Cédric Ollivier : +1 to Tom's sentence ;)
06:22:21 From Cédric Ollivier : Pierre: RC must be verified vs other CNTT compliant (regarding their testing scope) depl.
06:22:38 From Cédric Ollivier : Pierre: else it's chicken & egg
06:22:48 From Cédric Ollivier : (we know that very well in OPNFV)
06:23:54 From Cédric Ollivier : See Functest Gates
06:24:10 From Cédric Ollivier : vsperf gates (under construction?)
06:24:44 From Pankaj Goyal : Tom, does RI "rely" on RC other than for "conformance" (verify that it conforms to the RA)?
06:26:21 From Cédric Ollivier : Ideally RI should be fully verified by automated testing (RI->RC->RA). If the requirement is uncovered, it's somehow direct to RA (manual).
06:27:26 From Tom Kivlin : Pankaj - no, RI only relies on RC to verify that it is conforming to RA specs. So comes after RC in terms of development.
06:28:01 From Cédric Ollivier : +1 to Tom's sentence ;)
06:28:15 From Pierre Lynch : That's how we work it at ETSI NFV TST. We have (an equivalent to) RA, then once that is ready (OpenAPIs), then we (TST) can develop the tests (RC equivalent). Obviously we don't have an RI
06:28:51 From Pierre Lynch : Once RA v 3.3.1 (for example) is ready, then we develop TST (RC) v 3.3.1, released with the same number, but a few months later
06:29:47 From Cédric Ollivier : RI would be your develop gates (possibly only mocking in case of ETSI NFV TST).
06:30:19 From Pierre Lynch : Maybe. But ETSI doesn't really have RI :)
06:33:41 From Ildiko Vancsa : Release management guide in OpenStack: https://docs.openstack.org/project-team-guide/release-management.html
06:34:04 From Rihab Banday : We have run into situations in the past where RI2 has been based on older version of k8s, i.e conforming to older versions of RA2 & RC2. It could be difficult for RI software to keep up with new releases of RA/RC artifacts
06:34:34 From Ildiko Vancsa : https://docs.openstack.org/contributors/common/releases.html
06:34:54 From Cédric Ollivier : For my understanding the big tent is not that successful
06:35:09 From Cédric Ollivier : tempest is rolling
06:36:04 From Pankaj Goyal : FYI: While RA's typically follow RM, there can be exceptions. RM is just starting to define Observability, LCM, etc. but RA1 has some needed elements already.
06:37:38 From Ildiko Vancsa : Big tent doesn't have much to do with the release models
06:38:29 From Ildiko Vancsa : The challenges with big tent was that we've been having a lot of projects and not everything was clearly fitting into the scope of OpenStack
06:39:28 From Cédric Ollivier : Agree. It's more review and core developper issues
06:41:18 From Jim Baker (LFN) : One perspective I'd like to introduce is the role of the Anuket members in the two-phase release process. I think it essential that the spec writers are ACTIVELY involved in the acceptance/approval of the test release. The model of "write a spec, throw it over the wall" does not leverage the talent on the Anuket team. Can we figure out a process that keeps BOTH spec and test writers engaged in each others work product?
06:41:21 From Cédric Ollivier : Neutron could have simply promoted core developpers rather than allow decreasing the review quality ;)
06:43:14 From Pankaj Goyal : Bullet two needs changes RI doesn't follow RC. We can create an RI based on RA but wouldn't know how well it conforms to the specs -- that is what RC achieves
06:43:18 From Ildiko Vancsa : Well, that's the project's business on how they do onboarding and maintaining leadership. So I let the team members to judge the situation. :) With that said, there's always room for improvement everywhere!
06:44:12 From Cédric Ollivier : It's one point. They are lots of very good practices in OpenStack :)
06:47:33 From Gergely Csatari : Just a bit of clarification to what we just discussed with Beth. OpenStack has a binary project status, the project is either an official OpenStack project or not (and it is decided by the OpenStack TC), however in OIF they have different states, like Confirmed and Pilot, but here it is not clear for me who decides on the state of the projects (my guess is the OIF board). Maybe Ildiko can correct.
06:49:13 From Cédric Ollivier : Same for RC
06:49:56 From Ildiko Vancsa : Yes, for top level projects (OpenStack, Airship, Kata Containers, etc) it is the OpenInfra Foundation Board of Directors who make the decision when a project gets accepted as confirmed project
06:50:12 From Ildiko Vancsa : From the stage of being pilot for some time first
06:50:26 From Gergely Csatari : Thanks for the clarification.
06:51:41 From Ildiko Vancsa : And as @Gergely said for OpenStack it is the TC (Technical Committee) who can accept a new project team to become part of OpenStack
06:53:00 From Cédric Ollivier : RC is a test case list + a playbook and somehow a traceability (which is also the test entries). It lists containers and test case names. All the software is included in the containers.
06:53:23 From acm : I was writing: , RC is a test case list, the SW implementation of the test cases, and a framework to run the tests, collect results, store the results for later access (public).
06:54:24 From Pankaj Goyal : 1 RC but multiple RI's are possible
06:54:33 From Cédric Ollivier : RC selects a common test execution framemork which helps storing the results.
06:54:42 From Cédric Ollivier : +1

Action Items

- [ ]