Shift to Release Train

Catherine Lefevre & TAC Team

June 23, 2020

THE LINUX FOUNDATION

Agenda

- The Challenges from LFN Projects
- Release Cadence in Other Open Source projects
- Shift to Release Train & Brainstorming

The Challenges from LFN Projects

- How can we release more frequently, smaller scope without impacting the SW quality?
- How can we change the waterfall-ish development model?
- How can we get enough details about candidates requirements prior the release?
- How to manage the release in case of people turn-over?
- How to handle the scope and maintain a balance between Usecase/Functional reqs (attractive) and Non functional reqs (less sexy)?

Release Cadence in Other Open Source projects

- Linux Kernel – No Milestone & Checklist, No Formal Release Planning, No Use Case
  - Every 2-3 months
  - Developed in a single git repository
  - All changes are reviewed in a public mailing list. When a patch is ready for submission it is merged by a "subproject" maintainer to his -next git tree.
  - Every release starts with a 2-weeks merge window. During this time subsystem maintainers issue pull request to Linus (or higher level maintainer) asking him to pull changes from his git tree to main one.
  - When this time passes Linus releases release candidate 1 release then enter bugfix stage. This means that there are two streams of patches submitted:
    - Bug Fixes which are intended to enter next release candidate release
    - Main development patches which are merged to -next tree for a next main release
  - After 6-9 weeks of bug fixing a new version of the Linux kernel is released and the whole process starts over.

Release Cadence in Other Open Source projects

- DPDK - No Milestone & Checklist, No Formal Release Planning
  - Modeled on the Linux Kernel development model.
  - Every 3 months.
  - All changes are reviewed in a public mailing list with the help of patchwork tool in a similar fashion to the linux kernel.
  - Right after the previous release is out the merge window for a new release is open. All **new features must** be introduced **within the merge window**.
  - Last month of every release is considered a bug fixing period and no new functionality will enter the tree within this period.
  - The process is described in more details here.

Release Cadence in Other Open Source projects

- Openstack* - Release Planning incl. Requirements definition
  - 6 months long
  - The release takes 27 weeks which are numbered as week before the targeted release (R - 1, R - 2 etc). It *starts* with the *identification of community-wide goals and working on specs*. The spec is approved by affected project PTL and as soon as it is approved, the implementation can be started.
  - The deadline for *adding new specs is* R-12. R-7 is the deadline for adding **patches** with features to the Master. After this point of time only bugfixes are accepted. **Last 4 weeks** are Release Candidates and are **dedicated for testing and bug fixing**. More detailed schedule can be found here.
  - Using lunchpad and storyboard to maintain their progress

(*) Openstack was reorganizing their release cycle model so information mentioned here may be out of the date or mixed.

Release Cadence in Other Open Source projects

› ONAP - Release Planning incl. Requirements definition

- 2 Official Major Releases per Year + 1-2 Maintenance Releases and possibility of Self-Release per project
- ONAP Milestones and their associated checklist:
  - M0 – Kick-Off of the Release (Gathering Candidates List from the Community) while the former release is under testing.
  - TSC Prioritization – Review of the Candidates List and Ranking
    - RANK #0 – Special GO - quick wins, fully covered by involved companies
    - RANK #1 – TSC Must Have – Mandatory for the release
    - RANK #2 – Continuity - Items continued from previous releases
    - RANK #3 – PTL Go – items that PTLs is OK to include since team has bandwidth
    - RANK #4 – NO GO OR Not yet Reviewed by TSC – items not approved for various reasons
  - M1 – Release Plan – Commitments from the Companies/PTLS in alignment with TSC Prioritization
  - M2/M3 – Functionality/API Freeze
  - ***M4 - Development completion***
  - RC0 - Automated Pair-wise testing completion
  - RC1 - Final Dockers
  - RC2 - Completion of E2E Automated Testing
  - Sign-Off
  - Marketing Launch

ONAP Task Force Proposal

➢ Proposal

- 3 Releases per Year
- Main requirements artifacts are: Specifications, Features and Global Requirements
- Split the set of work items into Global Requirements set by the TSC and Specifications and Features from the projects and use case teams that are set by the working teams, including cross project commitments.
- Global Requirements are specific to a release.
- Specifications and Features can span across multiple releases and can scheduled to be "available" once all items are complete and tested.

Task Force Leads: Krzysztof Opasiak, Chaker Al-Hakim, Liam Fallon
More details on June 25th, 2020 @ 11am UTC – "Release cadence transition proposal"
https://zoom.us/j/98135653372

Brainstorming: Shift to Release Train

Release Scope proposal:
20 reqs
Tasks, Features fitting to a Sprint is the key
Code remains "Dark" until certification

Release scope achieved:
11 per project
Considering 3 RCs
9 leftovers, candidates for the next release

Reqs Owners, Subcommittees and TSC should establish the potential scope for the release at least 2 months in advance. At M1, Resources committed to PTLs. ***Scope should be decomposed considering dependencies*** *on other components;* ***Small dev. to fit to Dev Sprint Cycle****. Final scope is RC1+RC2+RC3 whatever DEV&TEST can deliver – PTL decide what it is possible*