

Release Lifecycle

Note: This page is out of date. See the current release process description here: <https://wiki.onap.org/x/iECLBQ>

The activities of the ONAP community are articulated around Projects and Releases.

The Release Lifecycle should be considered as a sub process of the overarching Project Lifecycle (as defined in ONAP Charter).

The **Project** Lifecycle should be seen as a **long term** endeavor (1-2 years) whereas the **Release** Lifecycle has a **short term** goal (next 5-6 months).

ONAP is adopting a **simultaneous** delivery model, meaning that all contributing projects have to follow the cadence and intermediate milestones.

The ONAP current release plan is available [here](#).



Definitions

A project is:

- Long term endeavor setup to deliver features across multiple releases
- Single entity solely responsible for one or multiple repositories
- Has a defined scope
- Led by PTL and committers with expertise in the relevant areas

A project is **not**:

- Release plan
- Collection of unrelated items
- Broadly scoped without clear rationale
- Existing without repo(s)
- Single-release vehicle
- Scoped so broadly to require committers with different expertise

A Release is:

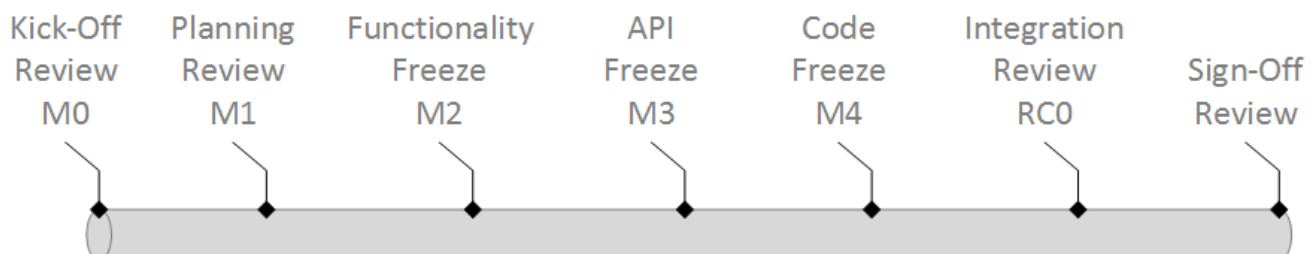
- Short term endeavor setup to deliver a specific Release Use case
- An agreed common framework (template, criteria, best practice) for all projects
- An unique release planning calendar with pre-defined milestones for each project to sync up with
- A vehicle to coordinate multiple projects and multiple type of projects (development, integration, packaging, deployment, documentation, training)
- Led by a Release Manager

Table of Contents

- 1 [Note: This page is out of date. See the current release process description here: <https://wiki.onap.org/x/iECLBQ>](#)
- 2 [Release Lifecycle Overview](#)
 - 2.1 [Release Kick-Off](#)
 - 2.2 [Release Planning](#)
 - 2.3 [Release Functionality Freeze](#)
 - 2.4 [Release API Freeze](#)
 - 2.5 [Release Code Freeze](#)
 - 2.6 [Release Candidate](#)
 - 2.7 [Release Sign-Off](#)
- 3 [Release Sub-Components](#)
- 4 [Release Reviews](#)
- 5 [Release Tailoring](#)

Release Lifecycle Overview

ONAP releases are organized through review milestones. For each milestone, every project has to fill out predefined template.



Release Kick-Off

Review	Milestone	Description	Activities
Kick-Off (Open Intent to Participate)	M0	<ul style="list-style-type: none"> The goal of the Release Kick-Off is to open for a potential Project its intention to participate in an ONAP Release. The proposed Project aligns with the architecture, team is in place, draft project plan is well-formed in wiki. Once approved by TSC, the planning process opens. Once approved, the team can start refining architecture, coding, testing, and documenting. Release Kick-Off review takes place for each next releases. TSC formally provides its formal disposition. The Release Kick-Off template includes: <ul style="list-style-type: none"> Project Description Scope Architecture alignment Define impacts on other projects Include estimated time and resource plans 	<p>To declare its intent to participate within a Release, the team must:</p> <ol style="list-style-type: none"> Establish Release Kick-Off Intent in wiki by filling out Release Kick-Off Template. Post Proposal in "Proposing a Project" wiki page. Inform ONAP TSC and Release Manager of their intention through the ONAP TSC mailing list. <p>Once the Release Kick-Off passed, the following activities are performed:</p> <ul style="list-style-type: none"> Release Use Case are iteratively worked on, submitted to TSC and proposed to GB. Development of feature code. Development of feature test cases (Functional Test Cases). Continuous Integration Continuous Delivery process is created (All repositories are created). The Integration Team defines an End to End Release Test Cases.

Release Planning

Review	Milestone	Description	Activities
Planning	M1	<ul style="list-style-type: none"> The goal of the Release Planning review is to ensure the plan is complete, sufficient, and aligned with release milestones. The team has a high level of confidence they will make the Release Sign-Off date. Issues brought to TSC. At Release Planning, the following activities have been achieved: <ul style="list-style-type: none"> Release Use Case(s) is (are) approved by GB. Release scope is defined and agreed on (Epics defined in Jira, Stories for next 2 Iterations groomed). All people resources are identified, documented and committed. All Release Components Name and structure is defined and documented. Project Plan supporting the Release is carefully documented. The functional Architecture for the project is defined and carefully documented (block diagram, sequence diagram,...) Dependencies between projects are identified. End 2 End Release Test Cases are defined (Integration Team). The vendors equipment list is defined. All OSs, VMs, VIMs, VNFs and hardware are defined. (Integration Team). The CI/CD Tool Chain is in place (Defined by Project Team and setup by LF). Documentation plan is available (Release Notes, Installation Guide, Configuration Guide, User and Admin Guide,...). Training Materials plan is available. The Installation strategy is defined. 	<p>To pass Release Planning review, each PTL must:</p> <ol style="list-style-type: none"> Fill out Release Planning Template and post it in project wiki Fill out Deliverables for Planning Milestones Template and post it in project wiki Inform ONAP TSC and Release Manager (through the ONAP TSC mailing list) prior to the milestone on the availability of the deliverable Have a concrete project time plan in place that accommodates dependencies and resource availability <p>Once the Release Planning passed, the team continues the architecture, coding, testing, documenting in an Agile 2 weeks iteration scrum framework.</p>

Release Functionality Freeze

Review	Milestone	Description	Activities
--------	-----------	-------------	------------

Functionality Freeze	M2	<ul style="list-style-type: none"> The goal of the Functionality Freeze is to mark the end of adding functionality in the Release. After Functionality Freeze, no new visible functionality is to be added to the current ONAP release. The API definition development activities are over and a stable document describing the API is documented and available to the community. At Functionality Freeze, the following activities have been achieved: <ul style="list-style-type: none"> All committed functionalities have been coded. All Functional Test Cases covering the scope of the release are defined (Project Team). All Functional Test Cases acceptance criteria are defined (Project Team). All source code has automated unit test (Project Team). The team is using the complete Linux Foundation environment (build, Jenkins, Gerrit, FOSS, Automated Unit Test, Nexus). A defined and documented final list of externally consumable APIs is available. All vendors equipments have been delivered (Integration team) 	<p>To Pass Functionality Freeze, the PTL must:</p> <ol style="list-style-type: none"> Fill out in project space the Deliverables for Functionality Freeze Milestone Checklist Template Inform ONAP TSC and Release Manager (through the ONAP TSC mailing list) prior to the milestone on the availability of the deliverable <p>After Functionality Freeze is passed, the team focus on:</p> <ul style="list-style-type: none"> Test Cases: <ul style="list-style-type: none"> Automate and execute the Functional Test Cases. Prioritize defects and address at least all critical and blocking defects.
----------------------	----	---	---

Release API Freeze

Review	Milestone	Description	Activities
API Freeze	M3	<ul style="list-style-type: none"> The goal of the API Freeze is to ensure API and Data Model are Frozen. At API Freeze, API stubs must be implemented. All provisional APIs are at least functional if not yet fully tested. At API Freeze, the following activities have been achieved: <ul style="list-style-type: none"> All externally accessible APIs & data models may not be modified. An API exception process will allow for critical changes to APIs after API Freeze. Any Changes to the API must be brought to the knowledge of the TSC for review and approval. APIs include, but are not limited to, all Java classes/interfaces declared public, all YANG models, all TOSCA profiles, all config file YANG schemas, and all REST/RESTCONF calls including the full URL with options. 50% of Functional Test Cases are automated (Project Team). Issues brought to TSC or Architecture Coordinator. 	<p>Prior to API review, Project teams must also review APIs Architecture Sub-committee.</p> <p>To pass API Freeze, the PTL must:</p> <ol style="list-style-type: none"> Fill out in project space the Deliverables for API Freeze Milestone Checklist Template template Inform ONAP TSC and Release Manager (through the ONAP TSC mailing list) prior to the milestone on the availability of the deliverable

Release Code Freeze

Review	Milestone	Description	Activities
Code Freeze	M4	<ul style="list-style-type: none"> The goal of the Code Freeze is to mark the end of the Features coding and testing and the resolution of all impacting defects (at least all Critical and Blocking defects) After Code Freeze, no new features/functionality are to be allowed into the current release. Only defects identified in the Jira system are allowed. The exceptions to this include new tests, and documentation. Defects found after Code Freeze are still defects and they may be created and worked on. This includes packaging and installation defects found as well. At Code Freeze, the following activities have been achieved: <ul style="list-style-type: none"> End 2 End Release Test Cases are automated (Integration Team). 100% of Functional Test Cases are automated (Project Team). The Labs are ready for ONAP testing. All OSs, VMs, VIMs, VNFs and hardware are installed and tested (Integration Team). Installation guide for each deliverables are available for review. 	<p>To pass Code Freeze, the PTL must:</p> <ol style="list-style-type: none"> Fill out in their Project Space the Deliverables for Code Freeze Milestone Template. Inform the TSC Chair and the Release Manager prior to the milestone on the availability of the deliverable. <p>After Code Freeze is passed, the team focus on:</p> <ul style="list-style-type: none"> Integration Testing <ul style="list-style-type: none"> Coordinate with central Integration Team to ensure all Functional Test Cases are stitched.

Release Candidate

Review	Milestones	Description	Activities
--------	------------	-------------	------------

Release Candidate	RC0, RC1, RC2	<ul style="list-style-type: none"> The goal of the Integration review is to ensure proper alignment and execution on: <ul style="list-style-type: none"> a. End 2 End Functional Test Cases (stitched test cases) integration testing for all projects participating in an ONAP release. b. End 2 End Release Test Cases. At that point, a fully-built, complete version of the current ONAP release is available. This includes all code, packaging and deployment that make up the final user-deliverable set of artifacts. Integration plan is further reviewed to ensure it covers correct API calls and data models between projects, alignment with use case and test framework. After RC0, new RCs will be continually built, e.g., once per day, to enable rapid testing and fixing of defects. Issues brought to TSC. At RC1 the Training Materials are available for review 	<p>To pass Integration review, the PTL must:</p> <ol style="list-style-type: none"> Fill out in their Project Space the Deliverables for RCx Milestone Template Inform the TSC Chair and the Release Manager prior to the milestone on the availability of the deliverable <p>After Integration review is passed, the team focus on:</p> <ul style="list-style-type: none"> Executing the Integration Plan. Prioritizing new defects and address them. In case there are blocking defects preventing the team to execute according to the plan, the defect must be fixed and a new full build made available. Reviewing and polishing documentation. Prepare activities to reach Release Sign-Off.
-------------------	---------------	--	--

Release Sign-Off

Review	Milestone	Description	Activities
Sign-Off		<ul style="list-style-type: none"> The goal of the Release Sign-Off review is to ensure that: <ul style="list-style-type: none"> The project has successfully passed all previous reviews. All committed deliverables are available and reached expected quality criteria. All testing activities are complete and successful. All documentation activities are complete and available. All Training Materials are complete and available. All blocking issues have been successfully addressed (otherwise there is no reason to hold the review). These above statements are the conditions for the release to be included in the ONAP target release. TSC formally provides its disposition. 	<p>To pass Release Sign-Off review, the team must:</p> <ol style="list-style-type: none"> Fill out in their Project Space the Deliverables for Release Sign-Off Milestone Checklist Template Inform the TSC Chair and the Release Manager prior to the milestone on the availability of the deliverables

Release Sub-Components

Some project teams may need to embed within their releases a set of Sub-Components. Sub-Components can be seen as a distinct piece of code that can be easily upgraded independently from the release deliverable. These Sub-Components must be clearly identified at Release Planning.

Release Components and Sub-Components are defined for all ONAP projects in a central place.

Some tailoring may take place to facilitate operation as we do not want to overload TSC voting members for relatively small sub-project updates.

Release Reviews

To avoid too much of formal review, TSC will provide its formal approval for project Kick-Off and Sign-Off only. However, all reviews must be carefully documented in project wiki.

The same principles applies for Release Reviews as for Project Reviews.

Release Tailoring

Depending on the release scope and project state, the Release Kick-Off and Release Planning Reviews may take place at the same time.

Architecture Review may be skipped for testing or documentation project.

Sub-Component Patch: when it is required to deliver a patch to address a critical issue, a simple email notification to TSC and Release Manager is required.