

Eclipse Foundation Specification Process

Table of Contents

Applicable Documents and Processes	2
Terms and Definitions	3
Structure and Organization	7
Specification Projects	7
Specifications	7
Committers	8
Specification Committee	9
Release Plans	9
Specification Process	11
Specification Project Lifecycle	11
Specification Version Lifecycle	12
Reviews	12
Ratification	14
Exceptions	16
Addendum: Process Revisions	17
Eclipse Development Process	17
Eclipse Foundation Specification Process	17
Working Group Specification Process	17
History	18
ChangeLog	18

DRAFT Version 1.2. Effective June 18, 2019

The document describes the Eclipse Foundation Specification Process (EFSP) for optional use by Eclipse Foundation Working Groups.

The EFSP leverages and augments the [Eclipse Development Process](#) (EDP). The EDP defines important concepts, including the Open Source Rules of Engagement, the organizational framework for open source projects and teams, releases, reviews, and more.

Although many of the activities related to this process are conducted by open source projects operating under the EDP, this specification process, and the Specification Versions delivered under it, are to be managed by Working Groups.

Subject to the approval of the Eclipse Management Organization (Executive Director or Delegate), individual Specification Committees may tailor the process for their unique requirements.

This document, and future revisions thereof will be approved by the Eclipse Foundation Board of Directors.

Applicable Documents and Processes

- [Eclipse Foundation Bylaws](#)
- [Working Group Process](#)
- [Working Group Participation Agreement](#)
- [Eclipse Foundation Specification License](#)
- [Eclipse Foundation TCK License](#)
- [Eclipse Foundation Trademark License Agreement](#)
- [Eclipse Foundation Membership Agreement](#)
- [Eclipse Development Process](#)
- [Eclipse Foundation Intellectual Property Policy](#)
- [Eclipse Foundation Anti-Trust Policy](#)

In the event of any conflict between the terms set forth in this EFSP and the terms of the documents listed above, the terms of those documents shall take precedence.

Terms and Definitions

Brand

The name and logo selected by the Working Group solely for the use of Compatible Implementations of Specifications designated by a Specification Committee.

Check Point Reviews

The Plan Review, the Progress Review, and the Release Reviews.

Committer

A developer who has the necessary rights to make decisions regarding a Project.

Compatible Implementation

Any implementation that fulfills all requirements of a Final Specification as demonstrated by fulfilling all requirements of the associated TCK.

Contribution

Content delivered to a Project under the terms of the [Eclipse Contributor Agreement](#).

Contributor

An individual who is a party to the [Eclipse Contributor Agreement](#).

Creation Review

A review to assess the community and membership response to a Project Proposal, verifies that appropriate resources are available for the project to achieve its plan, and serves as a Committer election for the project's initial Committers. For a complete definition, see the [EDP](#).

Final Specification

A Ratified Specification Version.

Individual Participant

An individual Committer on a Specification Project.

Major Release

A type of Release that includes either significant new functionality and/or breaking changes.

Member Participant

A Member of the Eclipse Foundation including Solutions Member, Enterprise Member, or Strategic Member (as defined in the [Eclipse Foundation Bylaws](#)) that has one or more Committers on a Specification Project.

Milestone Build

A build of the project content for limited distribution to demonstrate progress and solicit feedback.

Minor Release

A type of Release that includes new features over a Major Release.

Open Source License

One of the following OSI-approved open source licenses:

- [Eclipse Public License - v 2.0](#) (possibly with Secondary Licenses)
SPDX short identifier: EPL-2.0
- [Eclipse Distribution License - v 1.0](#)
SPDX short identifier: BSD-3-Clause
- [Apache License - v 2.0](#)
SPDX short identifier: Apache-2.0

This list may be modified with the unanimous approval of the Working Group Steering Committee and the Eclipse Foundation Board of Directors.

Participant

A Member Participant or Individual Participant.

Participant Representative

The Committer on a Specification Project who has the right to represent the interests (including without limitation the right to vote on behalf of) of a Participant. The Participant Representative of an Individual Participant is the same person.

Plan Review

A Review to approve a Release Plan to start a Release Cycle.

Pre-Proposal Phase

A phase in the Project lifecycle during which an individual or group of individuals declares their interest in, and rationale for, establishing a Project and assembles a proposal to create a new Specification Project. For a complete definition, see the [EDP](#).

Profile

A Specification that includes by reference a collection of Specifications and possibly additional requirements.

Progress Review

A type of Review that is used by a Project Team to summarize the accomplishments of the Project, verify that the Eclipse Development Process and IP Policy have been followed, and to highlight any remaining quality and/or architectural issues. For a complete definition, see the [EDP](#).

Project

A Project is the main operational unit by which all open source development occurs. For a complete definition, see the [EDP](#).

Project Management Committee (PMC)

The primary leadership of a Top-Level Project with responsibility to ensure that the Projects within its purview are active and viable. For a complete definition, see the [EDP](#).

Project Proposal

A document that describes the Project and the context in which the Project is being created. For more information, see the [EDP](#).

Project Leadership Chain

The leadership chain for a project is composed of the project's project lead(s), the leadership of the parent project (if any), the PMC leads and PMC members for the Top-Level Project, the EMO, and the EMO(ED). For a complete definition, see the [EDP](#).

Proposal Phase

A phase in the Project lifecycle during which a Project Proposal is presented to the community and Membership at Large to solicit feedback. For a complete definition, see the [EDP](#).

Ratified

A Specification Version that has been adopted by the Specification Committee and made available under the Eclipse Foundation Specification License to enable the creation and certification of Compatible Implementations.

Release

A Specification Version intended for ratification as a Final Specification.

Release Candidate

A feature-complete Milestone Build.

Release Cycle

The cycle of development that produces a Specification Version.

Release Plan

The description of activities to be undertaken as part of a Release Cycle to produce a Specification Version.

Release Review

A Release Review is a type of Progress Review that is aligned directly with a specific Release. This definition is the same as in the [EDP](#).

Review

The EFSP uses the same reviews as defined in the [EDP](#).

Scope

The defined scope of activities for a Specification Project.

Service Release

A Release that includes only minor changes and/or clarifications over a Major or Minor Release.

Specification

A collection of Application Programming Interface (API) definitions, descriptions of semantic behavior, data formats, protocols, and/or other referenced specifications, along with its TCK, intended to enable the development and testing of independent Compatible Implementations.

Specification Committee

A committee of a Working Group established to manage this Process for technologies within the scope of its Working Group.

Specification Document

The document that defines a Specification.

Specification Project

An Eclipse Foundation Project operating under the EDP and EFSP that is constituted to deliver Specification Versions.

Specification Team

The collective of Committers with responsibilities and privileges on a specific Specification Project.

Specification Version

A specific version of a Specification.

Super-majority

Two-thirds of the eligible voters.

Top-Level Project

An organizational unit that defines an overall mission and scope for a collection of Projects (and Specification Projects). For a complete definition, see the EDP.

Technology Compatibility Kit (TCK)

Software and documented requirements that support the testing of implementations to ensure that they are compatible with the Specification.

Termination of Participation

Occurs when an Individual Participant or Member Participant removes themselves or the Committers in their employ from a Specification Project.

Working Group

An Eclipse Foundation Working Group established under the [Eclipse Industry Working Group Process](#). Definitions from the Working Group Process are included herein by reference.

Other terms used in this document are defined in the [EDP](#).

Structure and Organization

A Specification Project is the main operational unit for Specification development at the Eclipse Foundation.

Specification Projects

Specification Projects operate under the supervision of both the Project Leadership Chain and the Specification Committee.

Among other things, the Scope of a Specification Project is intended to inform companies and individuals so they can determine whether or not to contribute to the Specification. Since a change in Scope may change the nature of the contribution to the project, a change to a Specification Project's Scope must be approved by a Super-majority of the Specification Committee.

Specifications

Specifications must be developed by Specification Projects.

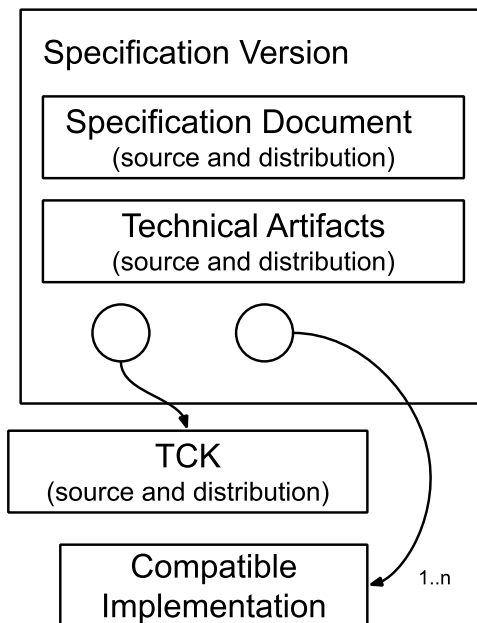
A Specification may describe parts as being optional. Optional parts of a Specification must not conflict with one another; it must be possible for a Compatible Implementation to implement all optional parts.

A Specification can define rules. If defined, such rules must not override the rules defined in any referenced Specification.

A Specification that aggregates other Specifications by reference may be designated as a Profile. Profiles do not have to be arranged in unique subsets (i.e. a Specification may appear in more than one Profile). A Super-majority, including a Super-majority of the Strategic Members of the Working Group, is required to approve a Profile Specification. A Specification Committee may, at its discretion, elect to label one or more Profiles as a "Platform".

Specification Versions

Each Specification Version references specific versions of its constituent artifacts. These artifacts include the Specification Documents, zero or more other Specifications, one or more Compatible Implementations licensed under an Open Source License, and exactly one associated TCK for this Specification.



Conceptual structure of a Specification Version

The Specification Document and related technical artifacts must be developed by the Specification Team.

Technology Compatibility Kits

There is exactly one TCK under an Open Source License for each Specification Version.

A specific version of a TCK is chosen by the Specification Project for each Specification Version; the TCK may be different for different Specification Versions.

Any implementation that fulfills all of the requirements of the TCK associated with a Final Specification may claim that it is a Compatible Implementation of that Final Specification. The TCK version associated with the Final Specification must not be modified other than as allowed or required by the rules of the TCK.

All parts of a Specification, including optional parts, should be covered by the TCK.

Compatible Implementations

A Compatible Implementation must fully implement all non-optional elements of a Specification Version, must not extend the API (no supersetting), and must fulfill all the requirements of the corresponding TCK. A Specification Version must identify at least one Compatible Implementation under an Open Source License that implements all optional elements of the Specification and fulfills the requirements of all elements (including optional elements) of the TCK.

Committers

Specification Project Committers must be Members of the Eclipse Foundation. Committers may be Members by virtue of working for a member organization, or may choose to complete the membership process independently.

All Specification Project Committers must be covered by a Working Group Participation Agreement.

Member Participants have the right to appoint a Participant Representative to every Specification Project that falls under the purview of the Specification Committee.

Specification Committee

The Specification Committee works with the PMC to manage the overall vision for the Specification Projects under their supervision.

Approvals

A Specification Committee must approve, by Super-majority, the following lifecycle events of Specification Projects:

- The creation of a new Specification Project;
- The Release Plan for a new Release Cycle of a Specification;
- Each revision to the Scope of a Specification;
- Each Review of a Specification Project, including the adoption of Specification Versions;
- A Profile Specification (this Super-majority must include a Super-majority of Strategic Members of the Working Group);
- A Platform designation (this Super-majority must include a Super-majority of Strategic Members of the Working Group); and
- Service Releases.

A ballot is used to seek Specification Committee approval. Unless otherwise stated in this process (or a Working Group-specific derivative of this process), the default period for all Specification Committee ballots is seven (7) days. During that time, any member of a Specification Committee may request that the period be extended to thirty (30) days.

A Specification Committee may opt to increase the length of the ballot period, but may not—under any circumstances—reduce any review period to fewer than seven (7) days.

All artifacts related to a ballot must be delivered in distribution form to the Specification Committee prior to the start of the ballot period, must not change during the ballot period (with the exception of minor corrections that do not change the semantic intent, as determined by the Specification Committee), and must persist in the delivered form following the ballot as part of the public record.

Release Plans

A Release Plan lists themes and areas of focus, describes Milestone Builds, and lists tentative dates for Reviews. The work defined by a Release Plan must be within the Scope of the Specification Project.

The exact requirements for a Release Plan, including the number and timing of Milestone Builds

and Reviews, are determined by the Project Leadership Chain and the Specification Committee. Minimally, a Release Plan must include a textual description of the activities planned for the Specification Version, and tentative dates for Milestone Builds, Progress Reviews, and the Release Review. Following approval, the Specification Committee must be notified of any changes to the dates of the Progress Review and the Release Review. The Specification Committee may, at their discretion, demand that the project team engage in additional Progress Reviews.

The Project Proposal serves as the Release Plan for the first release of a Specification Project.

A Release Plan must be approved by a Super-majority of the Specification Committee. If the Release Plan is rejected, the Specification Team may reapply at a future date.

Specification Process

The EFSP is based on the Development Process described in the [EDP](#).

Specification Project Lifecycle

The Specification [Project Lifecycle](#) is defined by the EDP.

Releases

While in the Incubation and Mature Phases, a Specification Project may engage in the Release process to produce Specification Versions which, when [Ratified](#), become Final Specifications. [Reviews](#) are required for all Releases of a Specification Project.

There are three types of Releases: Major, Minor, and Service. A Specification Team may consult with their PMC and Specification Committee to determine the appropriate classification.



The notions of Major, Minor, and Service Release bear a close resemblance to the "MAJOR.MINOR.PATCH" structure described by the Semantic Versioning specification. While a Specification Team may opt to use Semantic Versioning when naming their releases, this process imposes no requirement to do so. Further, this process imposes no requirement to tie any particular type of Release to any particular Semantic Versioning scheme.

Major and Minor Releases

A Major Release includes significant new features and/or breaking changes. A Minor Release includes new features over a Major Release.

A Specification Project's first Release must be a Major or Minor Release.

Milestone Builds

Leading up to a Release, a Specification Team must produce at least one Milestone Build. Milestone Builds and Release Candidates are "almost Releases" intended for adoption and testing by early-adopters. No formal Reviews are required for Milestone Builds.

Under no circumstances are Milestone Builds to be used as a substitute for doing proper official Releases.

All communication regarding Milestone Builds must include caveats explaining that these are not official Releases. Milestone Builds and Release Candidate builds must be labeled as such (e.g. **x.yMn**, **x.yRCn**, **alpha**, **beta**, or similar).

Service Releases

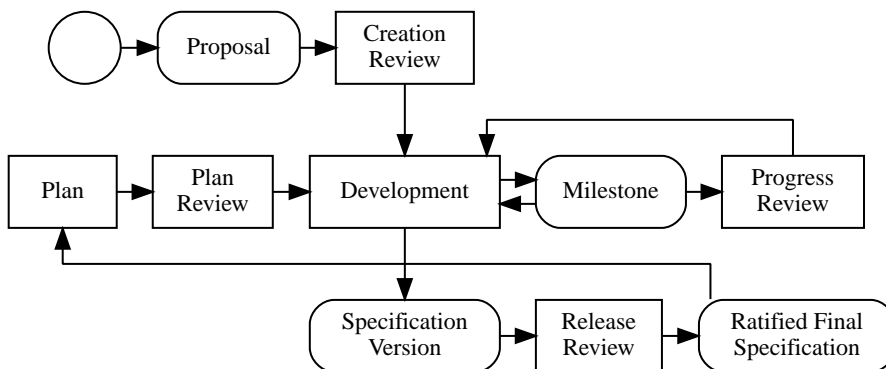
A Service Release includes only minor changes and/or clarifications over a Major or Minor Release. Specifically, a Service Release must not include any significant new features and/or breaking

changes. A Specification Team may consult with their PMC and Specification Committee to determine precisely what constitutes a minor change and/or clarification.

A Specification Team must have engaged in a successful Release Review for a Major or Minor Release prior to engaging in a Service Release. No Progress Review is required for a Service Release; the Specification Team must, however, engage in a successful [Release Review](#).

Specification Version Lifecycle

To produce a [Specification Version](#) (with the exception of [Service Releases](#)) a Specification Team must engage in a formal Release Cycle under the supervision of the Project Management Committee (PMC) and the Specification Committee.



An overview of the Eclipse Foundation Specification Process

A Specification Project's first Release Cycle starts with the successful completion of a Creation Review. To start a subsequent Release Cycle (and every Release Cycle thereafter), the Specification Team must present a Release Plan to the Specification Committee in a [Plan Review](#). The Plan Review must be approved by a Super-majority of the Specification Committee.

A Release Cycle ends when the Specification Team delivers a Specification Version to the EMO and Specification Committee via a [Release Review](#). To extend the Release Cycle, the Specification Team must stage a Milestone Build and engage in a [Progress Review](#). The Specification Team must engage in either a Release Review (which ends the Release Cycle) or a Progress Review (which extends the Release Cycle) within twelve (12) months of the last Review.

The Specification Team must deliver at least one Milestone Build to demonstrate progress and solicit feedback. Milestone Builds may be incomplete (for example, designated Compatible Implementations will not necessarily pass Milestone Builds of the TCK in their entirety). Subsequent Milestone Builds should, however, demonstrate progress. Later (feature complete) Milestone Builds may be referred to as *Release Candidates*. Milestones Builds should be staged for limited distribution to key stakeholders to solicit feedback.

The Specification Team must engage in a successful Release Review before the final Specification Version may be Ratified. A Specification Version becomes a Final Specification when it is Ratified.

Reviews

Reviews are a formal process through which all major lifecycle events and changes to Specification

Projects are announced and reviewed by the membership-at-large, and approved by the PMC, the Specification Committee, and the EMO.

A Specification Project may engage in all of the Reviews described by the EDP with the additional requirement that approval by a Super-majority of the Specification Committee is required to successfully complete a Review. Such Review shall include affirmation that the Specification Version in progress remains within the Scope of the Specification Project. Other additions and qualifications are noted in the descriptions of the reviews below.

Project Leads are responsible for initiating the appropriate Reviews. The Project Leadership Chain may also initiate a Review on the project's behalf.

The Specification Team must complete all required due diligence under the Eclipse IP Policy before initiating a Review.

Creation Review

Specification Projects are created using the process defined by the EDP with the added requirement that the Specification Committee must approve the Project Proposal by a Super-majority before the Specification Project can successfully complete a Creation Review.

The Specification Committee ballot and the Creation Review may run in parallel. The Project Proposal text must not be changed during the Creation Review period. If changes are required during this period, the Project Proposal is pushed back into the Proposal Phase.

Plan Review

A Plan Review provides a means for the Specification Team to present their Release Plan to the Project Leadership Chain, the Specification Committee, and the community for feedback. The Specification Committee must approve the Plan Review by a Super-majority ballot.

Progress Review

Progress Reviews are used to extend a [Release Cycle](#). During a Release Cycle a Project Team may be required to engage in one or more Progress Reviews.

The timing of a Progress Review must coincide with the staging of a Milestone Build which must be delivered to the PMC and the Specification Committee before the start of the Review.

The Specification Committee must approve the Progress Review by a Super-majority.

Progress Reviews may be combined with a Graduation or Restructuring Review, but must not be combined or overlap with a Release Review.

Release Review

A Specification Project must engage in a successful Release Review at the end of each Release Cycle.

The final build of the Specification Version's artifacts must be delivered to the PMC and Specification Committee before the start of the Release Review. The final build may be staged

before the start of the review, but must not be distributed as an official release until the Release Review is successfully completed.

The Specification Team must provide evidence that the TCK selected for the Specification Version provides sufficient coverage to reasonably validate Compatible Implementations.

The Specification Team must provide evidence that cited Compatible Implementations fulfill all requirements of the TCK and that at least one Compatible Implementation implements all optional aspects.

A Release Review concludes successfully with approval from the PMC and EMO, and approval by a Super-majority of the Specification Committee.

With approval, the Specification Project must release the final build of the artifacts of the Specification Version.

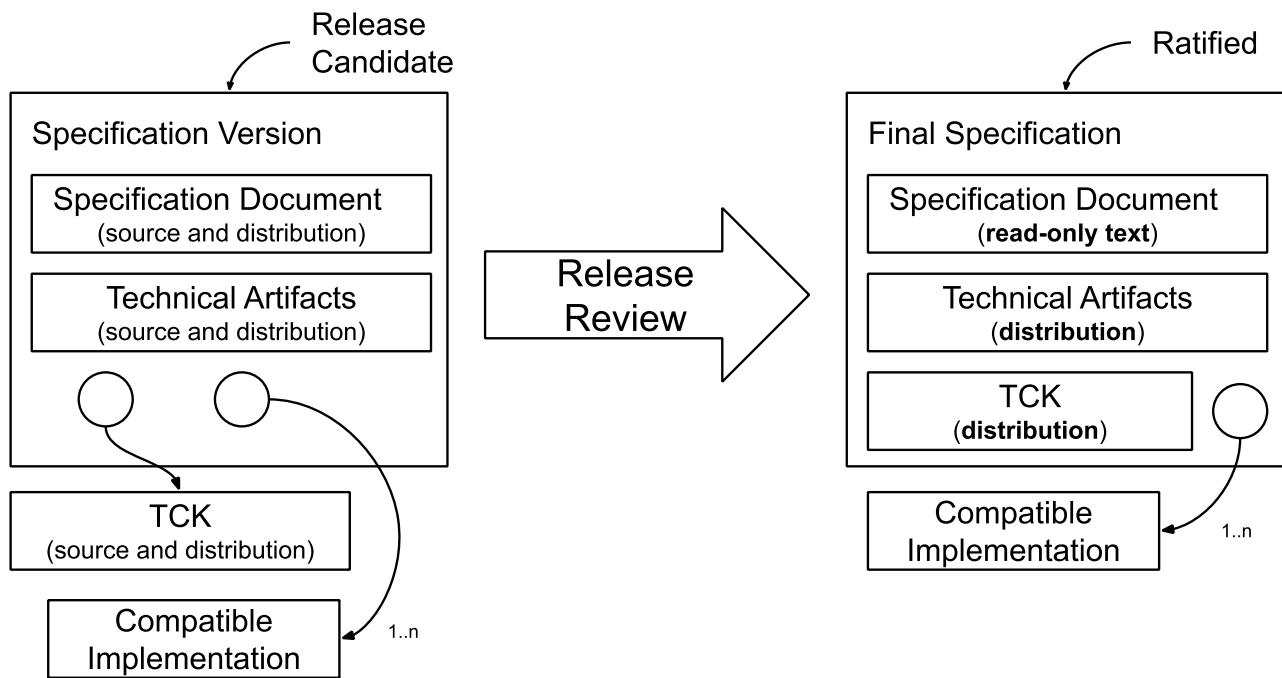
Ratification

With the approval of the Specification Committee by a Super-majority, a Specification Version is Ratified and the associated artifacts can be promoted and distributed by the Specification Committee as a Final Specification.

All Specification Versions referenced by a Ratified Final Specification must themselves be Ratified. The Release Review for prerequisite Specification Versions may be run concurrently with the Release Review for a referenced Specification Version.

The Specification Document for the Final Specification must be distributed as read-only text under the Eclipse Foundation Specification License. The Ratified TCK in composite must be distributed under the Eclipse Foundation Technology Compatibility Kit License. Other technical artifacts must be distributed under an Open Source License.

The diagram below is a conceptual model of the transition from a Specification Version to a Final Specification. No specific packaging technology or structure should be implied from this diagram.



Conceptual model of the transition from a Specification Version to a Final Specification. Note that no specific packaging technology or structure should be implied from this diagram.

Exceptions

Exceptions to this process may be granted only with Super-majority approval of the Specification Committee and approval of the Executive Director. All exceptions must be documented as part of the public record. Such documentation must include the exact change to the process, the case or conditions under which it applies, and the votes of each Specification Committee member.

Notwithstanding this exception process, no Specification Committee ballot period may be shorter than seven (7) days.

This exception process may not be used to override a Specification Committee member's request to extend the ballot period.

Addendum: Process Revisions

Eclipse Development Process

- Architecture Council initiates work, gathers requirements, and authors an updated document.
- Architecture Council approves final draft by simple majority (lazy consensus).
- Eclipse Board of Directors approves final draft by super-majority vote
- Version is published for use.

Eclipse Foundation Specification Process

- Executive Director convenes a committee (composed primarily, but not exclusively, of representatives from Working Group Specification Committees) and appoints a chairperson.
- Committee initiates work, gathers requirements, and authors an updated document
- Committee approves final draft by simple majority (lazy consensus).
- Eclipse Board of Directors approves the final draft.
- Version is published for use.



A Board delegation in place for the Executive Director to approve version 1.0 of the EFSP

Working Group Specification Process

- Working Group Specification Committee initiates work, gathers requirements, and authors an updated document.
- Working Group Specification Committee approves final draft by super-majority vote (as defined in the EFSP)
- Working Group Steering Committee approves final draft
- EMO(ED) approves final draft.
- Version is published for use.

History

Changes made in this document:

ChangeLog

[Version 1.2] - 2019-06-18

Added

- Clarification that all reviews are by default seven (7) days, see <https://github.com/EclipseFdn/EFSP/issues/14>.
- A section that describes how exceptions to the process may be granted, see <https://github.com/EclipseFdn/EFSP/issues/12>.

Changes

- Clarified that there must be exactly one TCK (not "TCK project") for each Specification Version.
- The requirement to engage in Progress Reviews is now time-based, see <https://github.com/EclipseFdn/EFSP/issues/13>.
- Reworded parts of the "Specification Version Lifecycle" section.
- Clarify that a formal Release Cycle is not required for a Service Release.
- Clarified that a Progress Review must not overlap with a Release Review.
- Recasted the defined term "Milestone" as "Milestone Build".

[Version 1.1] - 2019-03-20

Added

- Link to the Eclipse Foundation Specification License.
- Link to the Eclipse Foundation TCK License.
- Definitions of "Release", "Major Release", and "Minor Release".
- Service Releases require a Release Review.
- Specification Committee votes requires a Super-major of members of the Working Group (not members of the Eclipse Foundation).
- Specification Committee votes must be scheduled to run for a period of no less than one week.
- All artifacts related to a vote must be delivered in distribution form to the Specification Committee prior to the start of the vote, must not change during the voting period, and must persist in the delivered form following the vote as part of the public record.
- New section that describes releases.

Removed

- The Specification Project creation process diagram (moved to the FAQ).

Changes

- Simplified the definition of "Service Release".
- Made the delivery of Final Specifications inclusive of Service Releases.
- Use the term "ballot" rather than the ambiguous "vote".