

Eclipse Foundation Development Process Changes

This document highlights differences between this version of the specification process and the last. Additions are underlined and coloured green; deletions are ~~struck through and coloured red~~.

DRAFT This is based on a draft document that has not been approved.

You can find the draft document here ([./index.php](http://www.eclipse.org/development-process/)). Change are being tracked via GitLab (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/milestones/2#tab-issues>).

Version 1.11. ~~Effective September 12, 2023~~ **Planned** effective July 1, 2023 ~~2026~~.

Table of Contents

~~1. Purpose~~ 1. Purpose

~~1.1. Governance Documents~~

~~1.2. Terms and Definitions~~

~~2. Principles~~ 2. Principles

~~2.1 Open Source Rules of Engagement~~ 2.1. Open Source Rules of Engagement

~~2.2 Eclipse Ecosystem~~ 2.2. Three Communities

~~2.2.1. Developers~~

~~2.3 Three Communities~~

~~2.3.1 Developers~~ 2.2.2. Users

~~2.3.2 Users~~

~~2.3.3 Adopters~~ 2.2.3. Adopters

~~2.4 Clear, Concise, and Evolving~~ 2.3. Clear, Concise, and Evolving

~~2.5 Freedom of Action~~ 2.4. Freedom of Action

~~3. Requirements~~ 3. Requirements

~~3.1. Eclipse Foundation Responsibility~~

~~3.2. EMO Responsibility~~

~~4. Project Structure and Organization~~ 4. Project Structure and Organisation

~~4.1 Committers~~ 4.1. Committers

~~4.2 Code and Resources~~ 4.2. Code and Resources

~~4.3 Reserved~~ 4.3. Community Awareness

~~4.4 Community Awareness~~ 4.4. Scope

~~4.5 Scope~~ 4.5. Leaders

~~4.5.1. Project Management Committee (PMC)~~

~~4.6 Leaders~~

~~4.6.1 Project Management Committee (PMC)~~

~~4.6.2 Project Lead~~ 4.5.2. Project Lead

~~4.7 Committers and Contributors~~ 4.6. Committers and Contributors

~~4.8 [Reserved]~~ 4.7. Permanent Incubator Projects

4.9	Permanent Incubator Projects
4.10	Project Plans
4.8	Project Plans
5.	Reserved 5. Development Process
5.1.	Project Lifecycle
5.1.1.	Pre-Proposal Phase
6.	Development Process
6.1	Reserved 5.1.2. Proposal Phase
6.2	Project Lifecycle
6.2.1	Pre-Proposal Phase
6.2.2	Proposal Phase
6.2.3	Incubation Phase
6.2.4	Mature Phase
6.2.5	[Reserved]
6.2.6	Archived
5.1.5.	Archived
6.3	Reviews
5.2.	Reviews
6.3.1	Creation Review
5.2.1.	Creation Review
6.3.2	Graduation Review
5.2.2.	Graduation Review
6.3.3	Release Review
5.2.3.	Release Review
6.3.4	[Reserved]
5.2.4.	Progress Review
6.3.5	Progress Review
5.2.5.	Termination Review
6.3.6	Termination Review
5.2.6.	Restructuring Review
6.3.7	[Reserved]
6.3.8	Restructuring Review
6.3.9	Combining Reviews
5.2.7.	Combining Reviews
6.4	Releases
5.3.	Releases
6.5	Grievance Handling
6.	Grievance Handling
7.	Precedence
6.	Grievance Handling
8.	Revisions
7.	Precedence
8.	Revisions
8.1	[Reserved]
8.2	History
	ChangeLog
8.1.	ChangeLog

~~1. Purpose~~

1. Purpose

This document describes the development process for the Eclipse Foundation. In particular, it describes how the ~~Membership at Large~~, ~~General Assembly~~, the Board of Directors, other constituents of the Ecosystem, and the Eclipse Management Organization (EMO) lead, influence, and collaborate with Eclipse Projects to achieve the Purposes defined in the Bylaws (https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf).

This document has the following sections:

- ~~Principles~~Principles outlines the basic principles upon which the development process is based.
- ~~Requirements~~Requirements describes the requirements that the Eclipse community has for its development process.
- ~~Structure and Organization~~Project Structure and Organisation specifies the structure and

~~organization~~organisation of the Projects and Project community at Eclipse.

- ~~Development Process~~Development Process outlines the lifecycle and processes required of all Eclipse Projects.

This document defines terms used elsewhere in Eclipse governance documents. In the event of any conflict between the terms set forth in this document and the terms of the Governance Documents, listed below, the terms of those documents shall take precedence.

1.1. Governance Documents

All Committers and Project Leads engaged in Project activity are required to implement the policies and processes described by the Eclipse Foundations Governance Documents (<https://www.eclipse.org/org/documents/>), including:

- Eclipse Foundation Bylaws (https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf)
- Eclipse Foundation Membership Agreement (https://www.eclipse.org/org/documents/eclipse_membership_agreement.pdf)
- ~~{committerGuidelinesUrl}~~[Eclipse Committer Due Diligence Guidelines]
- Eclipse Foundation Intellectual Property Policy (https://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf)
- Eclipse Foundation Security Policy (<https://www.eclipse.org/security/policy.php>)
- Eclipse Foundation Antitrust Policy (https://www.eclipse.org/org/documents/Eclipse_Antitrust_Policy.pdf)
- Eclipse Foundation Community Code of Conduct (https://www.eclipse.org/org/documents/Community_Code_of_Conduct.php)

1.2. Terms and Definitions

Adopter :

Adopters are the individuals and ~~organizations~~organisations that adopt Eclipse technology for inclusion in their own products and services.

Architecture Council:

~~The Eclipse Architecture Council (AC) is a council of experienced Eclipse committers responsible for identifying and tackling any issues that hinder continued technological success and innovation, widespread adoption, and future growth of Eclipse technology. The Eclipse Architecture Council is formally defined in section VII of the Bylaws of the Eclipse Foundation.~~

Board of Directors :

The business and technical affairs of the Eclipse Foundation are managed by or under the direction of the Eclipse Board of Directors.

Check Point Review :

Check Point Review is a category of Review that includes: Creation Review, Progress Review, Release Review, and Graduation Review.

Committer :

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

Contributor :

A Contributor is anybody who makes contributions to a Project.

Creation Review :

A Creation Review is a type of Review that marks a Project's transition from the Proposal Phase to the Incubation Phase.

Developer :

A Committer or Contributor.

Ecosystem :

An Ecosystem is a system in which companies, organizations, organisations, and individuals all work together for mutual benefit.

Eclipse Management Organization:

The Eclipse Management Organization (EMO) ~~consists~~ is the operational arm of the Eclipse Foundation, responsible for overseeing the day-to-day execution of the Eclipse Foundation staff, Development Process. It provides essential governance and administrative support, including intellectual property due diligence, security audits, and project life cycle management. By acting as a neutral mediator, the EMO ensures that all projects operate with the transparency, meritocracy, and "vendor-neutrality" required for a healthy open source ecosystem.

Eclipse Architecture Council. Technology :

The open source technologies, specifications, platforms, runtimes, frameworks, and tools developed by Eclipse Projects are collectively referred to as *Eclipse Technology*.

EMO(ED) :

The term EMO(ED) generally refers to the Executive Director of the Eclipse Foundation. When discussing matters pertaining to the Eclipse Foundation Development Process, EMO(ED) refers to the subset of the EMO consisting of the Executive Director and whomever he or she may delegate specific approval authority to.

General Assembly :

As defined in the Eclipse Foundation Bylaws (https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf), the General Assembly is the supreme body of the Eclipse Foundation, composed of Strategic and Contributing Members, which is responsible for ensuring the achievement of the foundation's purposes.

Graduation Review :

A Graduation Review is a type of Review that marks a Project's transition from the Incubation Phase to the Mature Phase.

Incubation Branding :

Incubation Branding is the branding that must be applied to all Projects in the Incubation Phase.

Incubation Phase :

The Incubation Phase is a Phase during which the Project Team develops the process, the community, and the technology for a new Project.

Integration Build :

An Integration Build is a build of the Project content that is generated periodically and made available to the Developer community for testing purposes.

Maintainer :

A synonym for Committer.

Major Release :

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

Mature Phase :

The Mature Phase is a Phase during which a Project operates with an open and transparent process, has an actively involved and growing community, and produces Eclipse-quality technology.

Membership at Large Member :

~~The~~An Eclipse Foundation Member ("Member") is any individual, organisation, or affiliate that has executed an Eclipse Foundation Membership at Large (or *Membership*) refers to all members of all types, as defined by the Bylaws of the Eclipse Foundation (https://www.eclipse.org/org/documents/eclipse_foundation-bylaws.pdf).Agreement.

Milestone :

A Milestone is a build of the Project content intended for limited distribution to demonstrate progress and solicit feedback.

Minor Release :

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

Nightly Build :

A Nightly Build is a build of the Project content that is generated nightly and made available to the Developer community for testing purposes.

Permanent Incubator :

A Permanent Incubator is a Project that is intended to perpetually remain in the Incubation Phase.

Phase :

A Phase is a stage in the Project lifecycle. See Pre-proposal Phase, Proposal Phase, Incubation Phase, and Mature Phase.

Pre-proposal Phase :

The Pre-proposal Phase is a Phase during which a new Project Proposal is created.

Progress Review :

A Progress Review is a type of Review that is used by a Project Team to ~~summarizes~~summarise the accomplishments of the Project, verify that the Eclipse Foundation Development Process and IP Policy have been followed, and to highlight any remaining quality and/or architectural issues.

Project :

A Project is the main operational unit for open source software development at the Eclipse Foundation.

Project Lead :

A Project Lead, the first level in the Project Leadership Chain, is responsible for the overall well-being of a specific Project and serves as the primary liaison between that Project and the EMO.

Project Leadership Chain :

The Project Leadership Chain is composed of a 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).

Project Management Committee :

A Project Management Committee (PMC) is the primary leadership of a Top-Level Project with responsibility to ensure that the Projects within its purview are active and viable.

Project Proposal :

A Project Proposal (or just *Proposal*) is a document that is presented to the PMC, Membership at Large, General Assembly, and EMO to describe a potential new Project.

Project Team :

A Project Team is the collective of Committers with responsibilities and privileges on a specific Project.

Project Security Team :

The Project Security Team is a collective responsible for implementing the Eclipse Foundation Security Policy (<https://www.eclipse.org/security/policy.php>) in a specific Project.

Proposal Phase :

The Proposal Phase is a Phase during which a Project Proposal is presented to the community and Membership at Large General Assembly to solicit feedback.

Release :

A Release is a collection of Project artifacts intended for distribution beyond the Project Developers.

Release Candidate :

A Release Candidate is a feature-complete Milestone.

Release Review :

A Release Review is a type of Progress Review that is aligned directly with a specific Release.

Restructuring Review :

A Restructuring review is a type of Review that is used to notify the community of significant changes to one or more Projects.

Review :

A Review is a formally designated period of time during which the Project Management Committee, the Membership at Large, General Assembly, and the EMO are given an opportunity to survey the current state of a Project, provide feedback, and validate that the Project is in good standing.

Scope :

The Scope is the defined range of activities to be undertaken by a Project. The Project Team must operate within the bounds defined by the Project's Scope.

Service Release :

A Service Release, or *Bug-fix Release* is a type of Release that includes no significant changes or additions over the base Release.

Subproject :

A synonym for Project.

Termination Review :

A Termination Review is a type of Review that provides a final opportunity for a Project Team, the Project Leadership Chain, and the Eclipse Membership General Assembly to consider the proposed archival of a Project.

Top-Level Project :

A Top-Level Project is an organizational/organisational unit that defines an overall mission and scope for a collection of Projects.

Top-Level Project Charter :

A Top-Level Project Charter describes the mission, purpose, scope, and operational rules for a Top-Level Project.

2. Principles

2. Principles

The following describes the guiding principles used in developing this development process.

2.1 Open Source Rules of Engagement

2.1. Open Source Rules of Engagement

- **Open** - Eclipse is open to all; Eclipse provides the same opportunity to all. Everyone participates with the same rules; there are no rules to exclude any potential Contributors which include, of course, direct competitors in the marketplace.
- **Transparent** - Project discussions, minutes, deliberations, Project plans, plans for new features, and other artifacts are open, public, and easily accessible.
- **Meritocratic** - Eclipse is a meritocracy. The more you contribute the more responsibility you will earn. Leadership roles in Eclipse are also merit-based and earned by peer acclaim.

2.2 Eclipse Ecosystem

Eclipse as a brand is the sum of its parts (all of the Projects), and Projects should strive for the highest possible quality in extensible frameworks, exemplary applications, transparent processes, and Project openness.

The Eclipse Foundation has the responsibility to *...cultivate...an Ecosystem of complementary products, capabilities, and services...*. It is therefore a key principle that the Eclipse Foundation Development Process ensures that the Projects are managed for the benefit of both the open source community and the Ecosystem members. To this end, all Eclipse Projects are required to:

- communicate their Project plans and plans for new features (major and minor) in a timely, open and transparent manner;
- create high-quality frameworks capable of supporting the building of commercial grade products on top of them; and
- ship extensible, exemplary applications which help enable a broad community of users.

2.3 Three Communities

2.2. Three Communities

Essential to the Purposes of the Eclipse Foundation is the development of three inter-related communities around each Project:

2.3.1 Developers

2.2.1. Developers

A thriving, diverse, and active community of Developers is the key component of any Eclipse Project. Ideally, this community should be an open, transparent, inclusive, and diverse community of CommittersContributors and (non-Committer) Contributors.Committers. Attracting new Contributors and Committers to an open source Project is time consuming and requires active recruiting, not just passive "openness". The Project Leadership must make reasonable efforts to encourage and nurture promising new Contributors.

2.3.2 Users

2.2.2. Users

An active and engaged user community is proof-positive that the Project's exemplary applications areEclipse Technology is useful and needed. Furthermore, a large user community is one of the key factors in creating a viable Ecosystem around an Eclipse Project, thus encouraging additional open source and commercial organizationsorganisations to participate. Like all good things, a user community takes time

and effort to bring to fruition, but once established is typically self-sustaining.

~~2.3.3 Adopters~~

2.2.3. Adopters

An active and engaged Adopter community is the only way to prove that an Eclipse Project is providing ~~extensible frameworks and applications accessible via documented APIs. Reuse~~Eclipse Technology. Use of ~~the frameworks~~Eclipse Technology within the companies that are contributing to the Project is necessary, but not sufficient to demonstrate an Adopter community. Again, creating, encouraging, and nurturing an Adopter community outside of the Project's Developers takes time, energy, and creativity by the Project Leadership, but is essential to the Project's long-term open source success.

The Eclipse community considers the absence of any one or more of these communities as proof that the Project is not sufficiently open, transparent, and ~~inviting, and/or that it has emphasized applications at the expense of extensible frameworks or vice versa.~~inviting.

~~2.4 Clear, Concise, and Evolving~~

2.3. Clear, Concise, and Evolving

It is an explicit goal of the development process to be as clear and concise as possible so as to help the Project Teams navigate the complexities, avoid the pitfalls, and become successful as quickly as possible.

This document imposes requirements and constraints on the operation of the Projects, and it does so on behalf of the larger Eclipse community. It is an explicit goal of the development process to provide as much freedom and autonomy to the Projects as possible while ensuring the collective qualities benefit the entire Eclipse community.

Similarly, this document should not place undue constraints on the EMO or the Board of Directors that prevent them from governing the process as necessary. We cannot foresee all circumstances and as such should be cautious of being overly prescriptive and/or requiring certain fixed metrics.

The ~~frameworks, applications,~~Eclipse Technology, Projects, processes, community, and even the definition of quality continues to, and will continue to, evolve. Creating rules or processes that force a static snapshot of any of these is detrimental to the health, growth, and Ecosystem impact of Eclipse.

Part of the strength of this document is in what it does not say, and thus opens for community definition through convention, guidelines, and public consultation. A document with too much structure becomes too rigid and prevents the kind of innovation and change we desire for Eclipse. In areas where this document is vague, we expect the Projects and members to engage the community-at-large to clarify the current norms and expectations.

~~2.5 Freedom of Action~~

2.4. Freedom of Action

Projects are required to engage in practices that ensure the continued viability of the Project, independent from the continued availability of external resources and services, or continued participation on any single individual, ~~organization,~~organisation, or group.

In practical terms, Projects are required to use resources and services approved by the Eclipse Foundation. This includes (but is not limited to) all source code management, distribution channels for artifacts, issue tracking, documentation, and public communication channels.

~~3. Requirements~~

3. Requirements

This document is entirely composed of requirements. In addition to the requirements specified in this development process, the EMO is instructed to clarify, expand, and extend this process by creating a set of development guidelines to advance the creation, evolution, promotion, and support of the open source Projects; and to cultivate both a community and an Ecosystem of complementary products and services.

3.1. Eclipse Foundation Responsibility

The Eclipse Foundation acts as the open source steward for all Eclipse Projects by providing the legal framework, infrastructure, and governance required to ensure project technical integrity and regulatory compliance while facilitating transparent, vendor-neutral collaboration.

3.2. EMO Responsibility

The EMO has the responsibility and authority to mitigate issues that arise when Committers fail to perform the required ~~behaviors~~behaviours or engage in practices that risk harm to Eclipse Projects, the community, and/or Ecosystem. This includes, but is not limited to, issues that arise due to a failure to implement the Eclipse Foundation Security Policy (<https://www.eclipse.org/security/policy.php>), the Eclipse Foundation Intellectual Property Policy (https://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf), the Eclipse Foundation Community Code of Conduct (https://www.eclipse.org/org/documents/Community_Code_of_Conduct.php), or other governance policies of the Eclipse Foundation.

The EMO's authority includes, but is not limited to, the ability to grant specific individuals equivalent to Committer privileges, suspend access to Project resources, add or remove Committers and Project Leads, and in extreme cases terminate the Project.

The EMO is not permitted to override or ignore the requirements listed in this document without the express written endorsement of the Board of Directors.

The motivation for the EMO to take action along with description of the steps taken as part of the mitigation must be publicly disclosed within one week of action being taken. The ~~Grievance Handling~~Grievance Handling section of this document describes the means by which a Member may respond in the event that they believe that the EMO has overstepped its authority.

~~4. Project Structure and Organization~~

4. Project Structure and Organisation

A Project is the main operational unit at Eclipse. Specifically, all open source software development at Eclipse occurs within the context of a Project. Projects have leaders, Developers, code, builds, downloads, websites, and more. Projects are more than just the sum of their many parts, they are the means by which open source work is ~~organized~~organised when presented to the communities of Developers, ~~Adopters, Users,~~ and users. Adopters. Projects provide structure that helps Developers expose their hard work to a broad audience of consumers.

Eclipse Projects are ~~organized~~organised hierarchically. A special type of Project, *Top-Level Projects*, sit at the top of the hierarchy. Each Top-Level Project contains one or more Projects. Each Project may itself contain zero or more Projects. A Project that has one or more Projects is said to be the *parent* of those Projects. A Project that has a parent is oftentimes referred to as a *Subproject*. The term Project refers to

either a Top-Level Project or a Subproject.

The descendants of a Project are the Project itself and transitive closure of its child Projects. The top parent of a Project is the Top-Level Project at the top of the hierarchy.

Projects are the unit entity for:

- Committers;
- code and Releases;
- intellectual property (IP) records; and
- community awareness

4.1 Committers

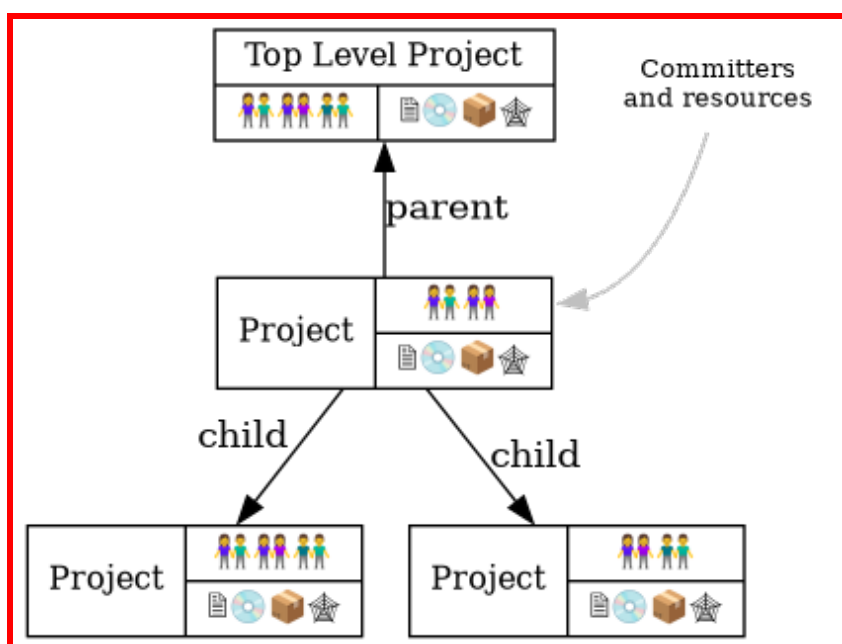
4.1. Committers

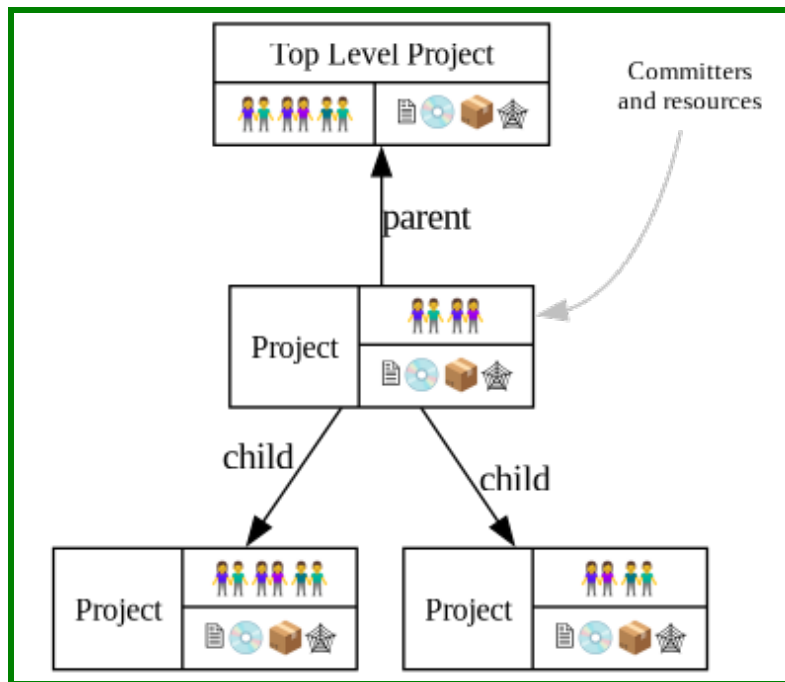
Each Project has exactly one set of Committers. Each Project's set of Committers is distinct from that of any other Project, including Subprojects or parent Projects. All Project Committers have equal rights and responsibilities within the Project. Partitioning of responsibility within a Project is managed using social convention. A Project may, for example, divide itself into logical partitions of functionality; it is social convention that prevents Committers from one logical partition from doing inappropriate work in another. If finer-grained management of Committer responsibilities is required, a Project should consider partitioning (via a Restructuring Review) into two or more Subprojects.

The Committers of a Project have the exclusive right to elect new Committers to their Project; no other group, including a parent Project, can force a Project to accept a new Committer.

There is no roll-up of Committers: the set of Committers on a Project is exactly that set of people who have been explicitly elected into that role for the Project (i.e. being a Committer on a Subproject does not give you any automatic rights on the "parent" Project or any child Project).

In practical terms, each Project has a single UNIX group Project Team of its Committers that provides are granted write-access to the Project's resources. Pictorially below, we see that a Project, in addition to the various resources and Committers it has, can also have zero or more Subprojects. Each of these Subprojects has its own distinct set Project Team of Committers and resources.





Project Committers, resources, and Subprojects

~~4.2 Code and Resources~~

4.2. Code and Resources

Each Project owns and maintains a collection of resources.

Resources may include source code, a Project website, space on the downloads server, access to build resources, and other services provided by the Eclipse Foundation infrastructure. The exact infrastructure provided by the Eclipse Foundation varies over time and is defined outside this process document.

A Project is not strictly required to make use of all the resources made available; a Project might, for example, opt to *not* maintain a source code repository. Such a Project might operate as an ~~organizational~~organisational unit, or container, for several Subprojects. Similarly, a Project might opt to provide a consolidated website, build and/or download site for its Subprojects (the Subprojects would then not require those resources for themselves).

Namespaces are assigned to a Project by the EMO. All Project source code must be ~~organized~~organised in the assigned namespaces and Projects can only Release code under their own namespace (that is, they cannot infringe on another Eclipse Project's namespace).

~~4.3 Reserved~~

~~4.4 Community Awareness~~

4.3. Community Awareness

Projects are the level of communication with the larger Eclipse community and Ecosystem. Projects may either have their own ~~communications (website, mailing lists, forums/newsgroups, etc)~~communication channels or they may be part of a parent Project's ~~communications (website, mailing list, forums/newsgroups, etc)~~communication channels. In either case, the Project is required to maintain an open and public communication channel with the Eclipse community including, but not limited to, Project plans, schedules, and design discussions.

All Projects must make the communication channels easy to find. Projects are further required to make the

separate communication channels of their child Projects (if any) easy to find.

~~Any Project in the Incubation Phase must correctly identify its website and Releases. A Project with at least one descendant Project in Incubation Phase must correctly annotate its own website so as to notify the Eclipse community that incubating Projects exist in its hierarchy. Any Release containing code from an Incubation Phase Project must be correctly labeled, i.e., the Incubation Phase is viral and expands to cover all Releases in which it is included.~~

~~4.5 Scope~~

4.4. Scope

Each Top-Level Project has a Charter which describes the purpose, Scope, and operational rules for the Top-Level Project. The charter should refer to, and describe any refinements to, the provisions of this development process. The Board of Directors approves the charter of each Top-Level Project.

Subprojects do not have separate charters; Subprojects operate under the charter of their parent Top-Level Project.

All Projects have a defined Scope and all initiatives within that Project are required to reside within that Scope. Initiatives and code that is found to be outside the Scope of a Project may result in the termination of the Project. The Scope of Top-Level Projects is part of the charter, as approved by the Board of Directors of the Eclipse Foundation.

The Scope of a Subproject is defined by the initial Project Proposal as Reviewed and approved by the Project Management Committee (PMC) (as further defined below) of the Project's top parent and by the EMO. A Project's Scope must be a subset of its parent's Scope.

~~4.6 Leaders~~

4.5. Leaders

There are two different types of Project Leadership at Eclipse: The Project Management Committee (PMC) and Project Leads. Both forms of leadership are required to:

- ensure that their Project is operating effectively by guiding the overall direction and by removing obstacles, solving problems, and resolving conflicts;
- operate using open source rules of engagement: meritocracy, transparency, and open participation; and
- ensure that the Project and its Subprojects (if any) conform to the Eclipse Foundation IP policy and procedures.

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~~Leads and PMC ~~members~~Members for the Top-Level Project, the EMO, and the EMO(ED).

In exceptional situations—such as Projects with zero active Committers, disruptive Committers, or no effective Project Leads—the Project Leadership Chain has the authority to make changes (add, remove) to the set of Committers and/or Project Leads of that Project, and otherwise act on behalf of the Project Lead.

~~4.6.1 Project Management Committee (PMC)~~

4.5.1. Project Management Committee (PMC)

Top-Level Projects are managed by a Project Management Committee (PMC). A PMC has one or more PMC ~~leads~~Leads and zero or more PMC Members. Together the PMC provides oversight and overall leadership for the Projects that fall under their Top-Level Project. The PMC as a whole, and the PMC ~~leads~~Leads in

particular, are ultimately responsible for ensuring that the Eclipse Foundation Development Process is understood and followed by their Projects. The PMC is additionally responsible for maintaining the Top-Level Project's charter.

PMC ~~leads~~Leads are approved by the Board of Directors; PMC ~~members~~Members are elected by the existing PMC ~~leads~~Leads and ~~members~~Members, and approved by the EMO(ED).

In the unlikely event that a ~~member of the~~ PMC Member becomes disruptive to the process or ceases to contribute for an extended period, the ~~member~~ PMC Member may be removed by the unanimous vote of the remaining PMC ~~members~~Leads and Members, subject to approval by the EMO. Removal of a PMC Lead requires approval of the Board of Directors.

4.6.2 Project Lead

4.5.2. Project Lead

Eclipse Projects are managed by one or more Project Leads. Project Leads are responsible for ensuring that their Project's Committers are following the Eclipse Foundation Development Process, and that the Project is engaging in the right sorts of activities to develop vibrant communities of ~~users, Adopters, Developers,~~ Users, and Contributors. ~~Adopters.~~ The initial Project Leads are appointed and approved in the Creation Review. Subsequently, additional Project Leads must be elected by the Project's Committers, approved by the Project's PMC, and appointed by the EMO(ED).

To hold a Project Lead role on a Project, an individual must also hold a Committer role on that Project.

In the unlikely event that a Project Lead becomes disruptive to the process or ceases to contribute for an extended period, the individual may be removed by the unanimous vote of the remaining Project Leads (if there are at least two other Project Leads), or unanimous vote of the Project's PMC.

4.7 Committers and Contributors

4.6. Committers and Contributors

Each Project has a development team, led by the Project Leaders. The development team is composed of Committers and Contributors. Contributors are individuals who contribute code, fixes, tests, documentation, or other work that is part of the Project. Committers have write access to the Project's ~~resources (source code repository, bug tracking system, website, build server, downloads, etc.)~~ resources and are expected to influence the Project's development.

Contributors who have the trust of the Project's Committers can, through election, be promoted Committer for that Project. The breadth of a Committer's influence corresponds to the breadth of their contribution. A development team's Contributors and Committers may (and should) come from a diverse set of ~~organizations.~~ organisations. A Committer gains voting rights allowing them to affect the future of the Project. Becoming a Committer is a privilege that is earned by contributing and showing discipline and good ~~judgment.~~ judgement. It is a responsibility that should be neither given nor taken lightly, nor is it a right based on employment by an Eclipse ~~member~~ Member company or any company employing existing Committers.

The election process begins with an existing Committer on the same Project nominating the Contributor. The Project's Committers will vote for a period of no less than one week. If there are at least three (3) positive votes and no negative votes within the voting period, the Contributor is recommended to the Project's PMC for commit privileges. If there are three (3) or fewer Committers on the Project, a unanimous positive vote of all Committers is substituted. If the PMC approves, and the Contributor signs the appropriate Committer legal agreements established by the EMO (wherein, at the very least, the Developer agrees to abide by the ~~Eclipse Intellectual Property Policy~~) Eclipse Committer Due Diligence Guidelines (<https://www.eclipse.org/legal/committer-guidelines>), the Contributor becomes a Committer and is given

write access to the source code for that Project.

At times, Committers may become inactive for a variety of reasons. The decision making process of the Project relies on active Committers who respond to discussions and vote in a constructive and timely manner. The Project Leads are responsible for ensuring the smooth operation of the Project. A Committer who is disruptive, does not participate actively, or has been inactive for an extended period may have his or her commit status revoked by the unanimous consent of the Project Leads. Unless otherwise specified, "an extended period" is defined as "no activity for more than six months".

Active participation in the user communication channels and the appropriate Developer mailing lists is a responsibility of all Committers, and is critical to the success of the Project. Committers are required to monitor and contribute to the user communication channels.

Committers are required to monitor the mailing lists associated with the Project. This is a condition of being granted commit rights to the Project. It is mandatory because Committers must participate in votes (which in some cases require a certain minimum number of votes) and must respond to the mailing list in a timely fashion in order to facilitate the smooth operation of the Project. When a Committer is granted commit rights they will be added to the appropriate mailing lists. A Committer must not be unsubscribed from a Developer mailing list unless their associated commit privileges are also revoked.

Committers are required to track, participate in, and vote on, relevant discussions in their associated Projects. There are three voting responses: +1 (yes), -1 (no, or veto), and 0 (abstain).

Committers are responsible for proactively reporting problems in the bug tracking system, and annotating problem reports with status information, explanations, clarifications, or requests for more information from Contributors. Committers are responsible for updating problem reports when they have done work related to the problem.

Committer, ~~PMC lead or member~~, Project Lead, and ~~council representative(s)~~ PMC Lead or Member are roles; an individual may take on more than one of these roles simultaneously.

~~4.8 [Reserved]~~

~~4.9 Permanent Incubator Projects~~

4.7. Permanent Incubator Projects

A Project may designate a Subproject as a "Permanent Incubator". A Permanent Incubator is a Project that is intended to perpetually remain in the ~~Incubation Phase~~ Incubation Phase. Permanent Incubators are an excellent place to innovate, test new ideas, grow functionality that may one day be moved into another Project, and develop new Committers.

Permanent Incubator Projects never have Releases. Permanent Incubators may have builds, and downloads. They conform to the standard Incubation Branding (<https://www.eclipse.org/projects/handbook/#incubation-branding>) requirements and are subject to the IP due diligence rules outlined for incubating Projects. Permanent Incubators do not graduate.

The Scope of a Permanent Incubator Project must fall within the Scope of its parent Project. The Committer group of the Permanent Incubator Project must overlap with that of the parent Project (at least one Committer from the parent Project must be a Committer for the incubator). The parent Project's Committers are responsible for ensuring that the incubator Project conforms to the rules set forth by the Eclipse Foundation Development Process.

A Permanent Incubator Project must be designated as such by including the word "Incubator" in its name (e.g. "Eclipse Dash Incubator"). To do otherwise is considered exceptional and requires approval from the

PMC and EMO(ED).

Only Top-Level Projects and Projects in the ~~Mature Phase~~Mature Phase may create a Permanent Incubator. Permanent Incubator Projects are created upon request; a Creation Review is not required.

~~4.10 Project Plans~~

4.8. Project Plans

Projects are required to make a Project plan available to their community at the beginning of the development cycle for each major and Minor Release. The plan may be as simple as a short description and a list of issues, or more detailed and complex. Subprojects may opt to include their plans with those of their parent Project.

Project Plans must be delivered to the community through communication channels approved by the EMO. The exact nature of the Project plan varies depending on numerous variables, including the size and expectations of the communities, and requirements specified by the PMC.

5. [Reserved]

6. Development Process

5. Development Process

Projects must work within their Scope. Projects that desire to expand beyond their current Scope must seek an enlargement of their Scope using a public Review as described below. Further, Projects must fit within the Scope defined by their containing Projects and the Scope defined in the charter of their Top-Level Project.

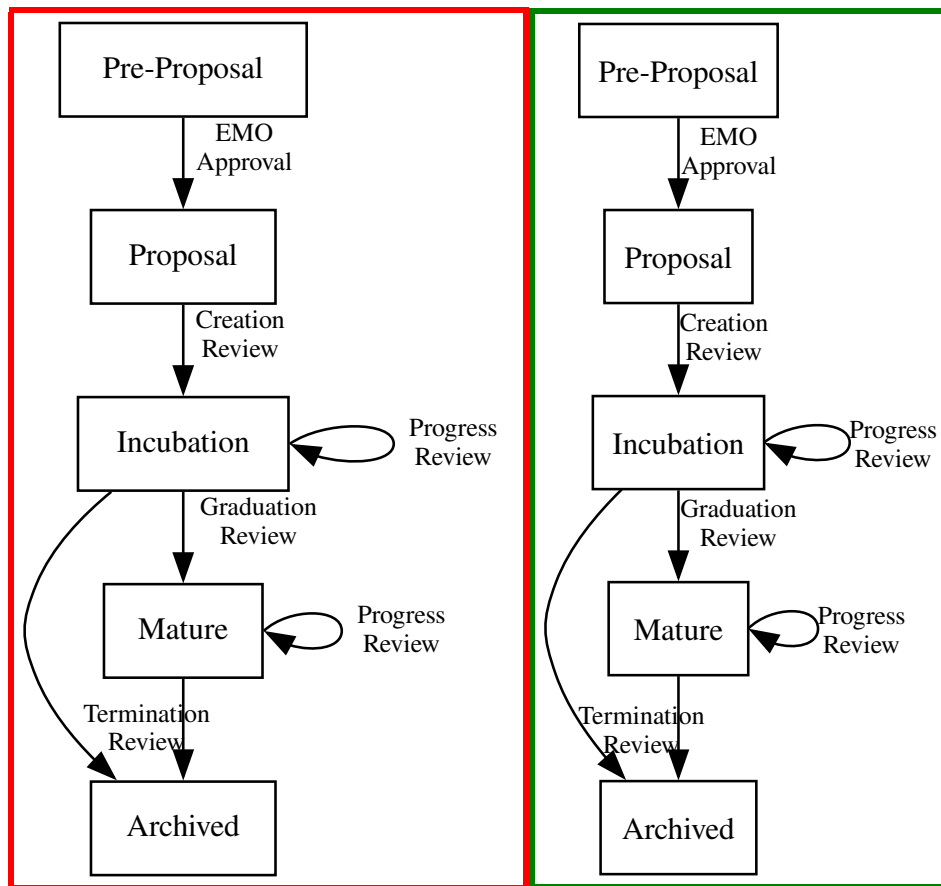
Projects must provide advanced notification of upcoming features via their Project plan.

~~6.1 Reserved~~

~~6.2 Project Lifecycle~~

5.1. Project Lifecycle

Projects go through distinct Phases. The transitions from Phase to Phase are open and transparent public Reviews.



An overview of the Project lifecycle Phases

6.2.1 Pre-Proposal Phase

5.1.1. Pre-Proposal Phase

An individual or group of individuals declares their interest in, and rationale for, establishing a Project. The EMO will assist such groups in the preparation of a Project Proposal.

The Pre-Proposal Phase ends when the proposal is published by EMO and announced to the ~~Membership at Large~~General Assembly by the EMO.

6.2.2 Proposal Phase

5.1.2. Proposal Phase

The proposers, in conjunction with the destination PMC and the community, collaborate in public to enhance, refine, and clarify the ~~proposal~~Project Proposal.

The Proposal Phase ends with a ~~Creation Review~~Creation Review, or withdrawal. The ~~proposal~~Project Proposal may be withdrawn by the proposers at any point before the start of a Creation Review. The EMO will withdraw a proposal that has been inactive for more than six months.

6.2.3 Incubation Phase

5.1.3. Incubation Phase

Moving a project to the Eclipse Foundation is a process, not an event. All Projects – with the exception to Top-Level Projects – start in the Incubation Phase, independent of code maturity or developer skill. New Projects require time to fully implement this process and correctly implement required policies. The Incubation Phase is a period of time during which a Project Team learns these processes and migrates their practices to align with the Open Source Rules of Engagement and corresponding policies and practices.

The Incubation Phase is an indicator to consumers that the Project may not fully implement practices that are consistent with this process. Projects in the Incubation Phase must annotate Project resources and

Release artifacts with Incubation Branding (<https://www.eclipse.org/projects/handbook/#incubation-branding>) as specified by the EMO.

The Incubation Phase is intended to be short. As a guideline, new Project teams should expect to spend three to six months in the Incubation Phase (longer periods are acceptable, but should be considered extraordinary). Projects transition from the Incubation Phase to the Mature Phase following a successful Graduation Review. Projects spend most of their life cycle in the Mature Phase.

The purpose of the Incubation Phase is to establish a fully-functioning open-open source Project. In this context, incubation is about developing the process, the community, and the technology. Incubation is a Phase rather than a place: new Projects may be incubated under any existing Project.

- A Project in the Incubation Phase can (and should) make Releases; Releases, engage in Committer and Project Lead elections, and other activities consistent with this process.
- Top-Level Projects skip incubation and are immediately put into the Mature Phase;
-

The Incubation Phase ends with ~~must not be used as a Graduation Review~~workaround or a Termination Review.

- Designated Permanent Incubator Projects remain perpetually in ~~excuse to avoid meeting the Incubation Phase; they do not create Releases, so no Reviews are required.~~

Many Eclipse Projects are proposed ~~obligations of this process~~ and initiated by individuals with extensive and successful software development experience. This document attempts to define a process that is sufficiently flexible to learn from all its participants. ~~At implementation of the same time, however, the Incubation Phase is useful for new Projects to learn the community-defined Eclipse-centric open source processes.~~required policies.

6.2.4 Mature Phase

5.1.4. Mature Phase

The Project Team has demonstrated that they are an open-open source Project with an open and transparent process, an actively involved and growing community, and Eclipse-quality technology. The Project is now a mature member of the Eclipse community.

6.2.5 [Reserved]

6.2.6 Archived

5.1.5. Archived

Projects that become inactive, either through dwindling resources or by reaching their natural conclusion, are archived. Projects are moved to archived status through a Termination Review.

If there is sufficient community interest in reactivating an archived Project, the Project can start again with a Creation Review. As there must be good reasons to have terminated a Project, the Creation Review provides a sufficiently high bar to prove that those reasons are no longer valid.

6.3 Reviews

5.2. Reviews

The Eclipse Foundation Development Process is predicated on open and transparent behavior. All major changes to Eclipse Projects must be announced and Reviewed by the Membership at Large. Major changes include the Project Phase transitions as well as the introduction or

~~exclusion~~removal of significant new technology or capability. It is a clear requirement of this document that members who are monitoring the appropriate media channels not be surprised by the post-facto actions of the Projects.

Projects are responsible for initiating the appropriate Reviews. If it is determined to be necessary, the Project Leadership Chain (e.g. the PMC or EMO) may initiate a Review on the Project's behalf.

All Reviews have the same general process:

1. The Project Team will complete all required due diligence under the Eclipse IP Policy prior to initiating the Review.
2. A Project representative (Project Lead or Committer) assembles Review documentation.
3. A Project representative presents the Review documentation to the Project's PMC along with a request to proceed with the Review and for approval of the corresponding documentation.
4. Upon receiving approval from the PMC, a Project representative makes a request to the EMO to schedule the Review.
5. The EMO announces the Review schedule and makes the documentation available to the ~~Membership at Large~~General Assembly.
6. The EMO approves or fails the Review based on the feedback, the Scope of the Project, and the Purposes of the Eclipse Foundation as defined in the bylaws.

The Review documentation requirements, and criteria for the successful completion of each type of Review will be documented by the EMO. PMCs may establish additional success criteria.

The Review period is open for no less than one week and usually no more than two weeks. The Review ends with the announcement of the results in the defined Review communication channel.

~~If any~~Any member who believes that the EMO has acted incorrectly in approving or failing a Review may appeal ~~to via~~ the ~~Board of Directors to Review the EMO's decision~~Grievance Handling process

~~6.3.1 Creation Review~~

~~5.2.1. Creation Review~~

The purpose of the Creation Review is to assess the community and ~~Membership at Large~~General Assembly's response to the ~~proposal~~Project Proposal, to verify that appropriate resources are available for the Project to achieve its plan, and to serve as a Committer election for the Project's initial Committers. The Eclipse Foundation strives not to be a repository of "code dumps" and thus Projects must be sufficiently staffed for forward progress.

~~The Creation Review documents must include short nomination bios of the proposed initial Committers. These bios should discuss their relationship to, and history with, the incoming code and/or their involvement with the area/technologies covered by the proposal. The goal is to help keep any legacy Contributors connected to new Project and explain that connection to the current and future Eclipse Membership at Large, as well as justify the initial Committers' participation in a meritocracy.~~

~~6.3.2 Graduation Review~~

~~5.2.2. Graduation Review~~

The purpose of the Graduation Review is to mark a Project's change from the Incubation Phase to the Mature Phase.

The Graduation Review confirms that the Project is/has:

- ~~A working and demonstrable code base of sufficiently high quality.~~
- Active and sufficiently diverse communities appropriate to the size of the graduating code base: ~~Adopters, Developers, Users, and users.~~Adopters.

- Operating fully in the open following the ~~principles and Purposes of Eclipse.Principles.~~
- A credit to Eclipse and is functioning well within the larger Eclipse community.

A Graduation Review is generally ~~combined~~combined with a ~~progress~~Progress Review or ~~Release Review.~~Release Review.

~~6.3.3 Release Review~~

5.2.3. Release Review

A Release Review is a type of ~~Progress Review~~Progress Review that is aligned directly with a specific Release. A Release Review must be concluded successfully before the corresponding ~~Release~~Release is announced to the community.

~~6.3.4 [Reserved]~~

~~6.3.5 Progress Review~~

5.2.4. Progress Review

The purposes of a Progress Review are: to ~~summarize~~summarise the accomplishments of the Project, to verify that the IP Policy is being followed, to highlight any remaining quality and/or architectural issues, and to verify that the Project is continuing to operate according to the ~~principles and Purposes of Eclipse.Principles.~~

~~6.3.6 Termination Review~~

5.2.5. Termination Review

The purpose of a Termination Review is to provide a final opportunity for the ~~Committers and/or Eclipse Membership at Large~~Committers, Project Management Committee, and General Assembly to discuss the proposed archiving of a Project. The desired outcome is to find sufficient evidence of renewed interest and resources in keeping the Project active.

~~6.3.7 [Reserved]~~

~~6.3.8 Restructuring Review~~

5.2.6. Restructuring Review

The purpose of a Restructuring Review is to notify the community of significant changes to one or more Projects. Examples of "significant changes" include:

- Movement of significant ~~chunks of functionality~~content from one Project to another.
- Modification of the Project structure, e.g. combining multiple Projects into a single Project, or decomposing a single Project into multiple Projects.
- Change of Project Scope.

~~6.3.9 Combining Reviews~~

5.2.7. Combining Reviews

Reviews can be combined at the discretion of the PMC and EMO. Multiple Projects may participate in a single Review. Similarly, multiple Review types can be engaged in simultaneously. A parent Project may, for example, engage in an aggregated Progress Review involving itself and some or all of its child Projects; a consolidated Restructuring Review may move the code for several Projects; or a Progress Review may be combined with a Graduation Review. When multiple Reviews are combined, the Review documentation must explicitly state all of the Projects and types of Reviews involved, and include the required information about each.

It should be noted that the purpose of combining Reviews is to better serve the community, rather than to

reduce effort on the part of the Project (though it is fortunate when it does both). Combining a progress and Graduation Review, or aggregating a Progress Review of a Project and several of its child Projects generally makes sense. Combining Progress Reviews for multiple unrelated Projects most likely does not.

6.4 Releases

5.3. Releases

Any Project, with exception of Permanent Incubators, may make a Release. A Release may include the code from any subset of the Project's descendants.

(Most of this section is paraphrased from the excellent Apache Software Foundation Releases FAQ (<http://www.apache.org/dev/release.html>). The Eclipse community has many of the same beliefs about Releases as does the Apache community and their words were already excellent. The Apache Software Foundation Releases FAQ is distributed under the Apache License, Version 2.0 (<http://www.apache.org/licenses/LICENSE-2.0>.)

Releases are, by definition, anything that is distributed outside of the Committers of a Project. If consumers are being directed to download a build, then that build has been released (modulo the exceptions below). All Projects and Committers must obey the Eclipse Foundation requirements on approving any Release.

The Eclipse IP Policy (https://www.eclipse.org/org/documents/Eclipse_IP_Policy.pdf) must be followed at all times. Prior to issuing any Release, the Project Lead must assert that the IP Policy has been followed and all approvals have been received.

The Project Team must provide (either directly or indirectly) a link between the distribution form of Release artifacts and the corresponding source code.

A Project can make official Releases for one calendar year following a successful Progress Review or Release Review. The Project Leadership Chain may, at its discretion, require that the Project engage in additional Reviews (e.g. a progress or Release Review) prior to issuing a Release.

Exception 1: Nightly and Integration Builds During the process of developing software and preparing a Release, various nightly and Integration Builds are made available to the Developer community for testing purposes. Do not include any links on the Project website, blogs, wikis, etc. that might encourage non-early-adopters to download and use Nightly Builds, Release Candidates, or any other similar package (links aimed at early-adopters and the Project's Developers are both permitted and encouraged). The only people who are supposed to know about such packages are the people following the Developer mailing list and thus are aware of the limitations of such builds.

Exception 2: Milestone and Release Candidate Builds Projects are encouraged to use an agile development process including regular Milestones (for example, six week Milestones). Milestones and Release Candidates are "almost Releases" intended for adoption and testing by early-adopters. Projects are allowed to have links on the Project website, blogs, wikis, etc. to encourage these outside the Committer circle early-adopters to download and test the Milestones and Release Candidates, but such communications must include caveats explaining that these are not official Releases. Milestones and Release Candidate builds must be labeled as such (e.g. `x.yMn`, `x.yRCn`, *alpha*, *beta*, or similar).

Exception 3: Service Releases with no new features Service Releases with no significant changes or additions over the base Release are allowed to be released without an additional Review.

Under no circumstances are builds and Milestones to be used as a substitute for doing proper official Releases. Proper Release management and Reviews is a key aspect of Eclipse quality.

Releases and corresponding artifacts for Projects in the Incubation Phase must be labeled labelled to

indicate the incubation status of the Project. See ~~Incubation Branding~~ (<https://www.eclipse.org/projects/handbook/#incubation-branding>) for more information.

~~6.5 Grievance Handling~~

6. Grievance Handling

When a member has a concern about a Project, the member will raise that concern with the Project's leadership. If the member is not satisfied with the result, the member can raise the concern with the parent Project's leadership. The member can continue appeals up the Project Leadership Chain and, if still not satisfied, thence to the EMO, then the executive director, and finally to the Board of Directors. All appeals and discussions will abide by the guiding principles of being open, transparent, and public.

Member concerns may include:

- Out of Scope. It is alleged that a Project is exceeding its approved Scope.
- Dysfunctional. It is alleged that a Project is not functioning correctly or is in violation of one or more requirements of the Eclipse Foundation Development Process.
- Contributor appeal. It is alleged that a Contributor who desires to be a Committer is not being treated fairly.
- Invalid veto. It is alleged that a -1 vote on a Review is not in the interests of the Project and/or of Eclipse.

A variety of grievance resolutions are available to the EMO up to, and including, rebooting or restarting a Project with new Committers and leadership.

~~7. Precedence~~

7. Precedence

In the event of a conflict between this document and a Board of Directors-approved Top-Level Project charter, the most recently approved document will take precedence.

~~8. Revisions~~

8. Revisions

As specified in the Bylaws, the EMO is responsible for maintaining this document and all changes must be approved by the Board of Directors.

Due to the continued evolution of the Eclipse technology, the Eclipse community, and the software marketplace, it is expected that the Eclipse Foundation Development Process (this document) will be reviewed and revised on at least an annual basis.

The EMO is further responsible for ensuring that all plans, documents and reports produced in accordance with this development process be made available to the ~~Membership at Large~~General Assembly via an appropriate mechanism in a timely, effective manner.

~~8.1 [Reserved]~~

8.1. ChangeLog

8.2 History

Changes made in this document:

ChangeLog[2026] - 2026-07-01

[2023] - 2023-09-01 Added

Added

- Eclipse Foundation is the open source steward for all Eclipse Projects #24 (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/issues/24>)
- Affirm that PMC Leads are required to vote in matters that pertain to the PMC #17 (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/issues/17>)
- Assert that the Incubation Phase is a time during which project teams may not get everything right, and that's okay.

Removed

- References to the Architecture Council #22 (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/issues/22>)
- Requirement that Project Proposal documents include "nomination bios" of initial committers

Changed

- References to the "Membership at Large" have been changed to "General Assembly"
- Definition of EMO has been changed to reflect recent changes to the Bylaws
- References to "extensible frameworks, exemplary applications" (and similar) changed to "Eclipse Technology" #25 (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/issues/25>)
- Mentions of specific communication technologies (websites, mailing lists, forums, ...) replaced with "communication channels"
- Minor: consistently refer to "Developers, Users, and Adopters"
- Releases are content that is intended to be distributed outside of the Committers of the Project; attempts to further box this with exceptions have been removed #23 (<https://gitlab.eclipse.org/eclipse/vhant/development-process/-/issues/23>)
- Grievance Handling for Reviews point to the standard Grievance Handling process
- British English applied

[2023] - 2023-09-01

Added

- List applicable governance documents #11 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/11>)
- Define EMO(ED) as a term
- Project leads must also be committers #9 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/9>)
- Clarification that the unanimous consent of the project leads is required to retire a committer

Removed

Removed

- All references to mentors #1 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/1>)
- References to the "coordinated release" #13 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/>)

policies/development-process/-/issues/13)

Changes

Changes

- Mentions of "Eclipse Development Process" changed to "Eclipse *Foundation* Development Process" #2 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/2>)
- "Meritocracy" changed to "Meritocratic" in the Open Source Rules of Engagement to be more consistent grammatically with "Open" and "Transparent"
- Fixed bogus references to the Bylaws. Specifically removed a quotation of an older version of the Purposes defined in the Bylaws. #6 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/6>)
- Added section 3.1 which extends the options available to the EMO to mitigate the situation when committers fail to perform required behaviours or engage in practices that risk harm to an Eclipse project. Previously, the only option available to the EMO was to terminate the project #12 (<https://gitlab.eclipse.org/eclipsefdn/emo-team/policies/development-process/-/issues/12>)

[2020] - 2020-11-01

[2020] - 2020-11-01Removed

Removed

- All references to the parallel IP Process
- The "Revision" section (formerly 8.1) that had indicated the board meeting in which the specific revision was approved (this is inconsistent with how we treat other documents).

Changes

Changes

- Document title changed to "Eclipse Foundation Development Process" to make it consistent with other documents.
- Section 4.7 renamed from "Committers" to "Committers and Contributors"
- In section 4.7, the reference to "submitters" changed to "Contributors"

[2020] - 2020-01-20

[2020] - 2020-01-20Removed

Removed

- All references to the Eclipse Planning Council have been removed
- The entire section 4.8, which quoted text from the bylaws regarding councils has been removed (the "Terms" section covers this)

Changes

Changes

- A reference to the bylaws was added in the definition of the Eclipse Architecture Council.

[2018] - 2018-12-14

[2018] - 2018-12-14Added

Added

- A separate section that formally defines key terms has been added.
- The notion of a Progress Review added to augment (and in some cases, replace) Release Reviews

(Bug 496321).

- Added a requirement that the Project Team must provide (either directly or indirectly) a link between the distribution form of Release artifacts and the corresponding source code.
- A CHANGELOG based on "keep a changelog" <https://keepachangelog.com/> (<https://keepachangelog.com/>)

Removed

Removed

- Removed redundant restriction that Permanent Incubators cannot participate in the simultaneous release (they can't release at all).
- Removed discussion regarding specific labeling of Milestones and Release Candidates (replaced with an simple statement that Milestones and Release Candidates must be labeled as such with examples)

Changes

Changes

- Converted document source format to AsciiDoctor.
- The time period for votes and reviews is expressed in terms of simple weeks rather than "one week of generally accepted business days" which is unnecessarily complicated.
- Removed the unnecessary statement that "Major releases continue to go through release reviews." from the description of the Mature Phase.
- Content in the Releases section has been reorganized.
- Release Reviews have been recast as a special type of Progress Review.
- The requirement that a project engage in a Release Review prior to a release has been changed to a requirement that project can only release within a year of successfully engaging in either a Progress Review or a Release Review. Further, the Project Leadership Chain is granted the ability to compel a project to engage in a Progress Review.
- The requirement that the "the IP Policy has been followed and all approvals have been received" has been moved from being a requirement of a Release Review to being a requirement of a Release. This is a bit subtle, but important given that with the introduction of a Progress Review a project may issue Releases without engaging in a Release Review.
- Content in the Releases section has been reorganized so that the three exceptions are grouped followed by the discussion regarding milestones.
- The notion "public comments" has been generalized to "feedback" in the Reviews section (Bug 538613).
- Change the title of section 2.3.1 to "Developers", and section 4.7 to "Committers" (Bug 484587).
- References to "Bug fix Releases" changed to "Service Releases" in the Releases section.

[2015] - 2015-12-31

[2015] - 2015-12-31 Changes

Changes

- Make the process of retiring/removing project members explicit (Bug 376001).
- Streamline review requirements (Bug 415620).
- Rewrite the section on requirements to be more clear and succinct (Bug 416185).
- Appointed members of the Architecture Council are appointed to two (Bug 418208) year renewable terms.
- Include "Freedom of Action" requirement in the EDP (Bug 471367).

- Generalize the expected output of projects (Bug 477238).
 - Include definitions of terms required by other legal documents (Bug 480526).
 - Only one mentor required for an incubating project (Bug 463850).
-

Projects ()

What's New? (/projects/whatsnew.php)

List of Projects (/projects/listofprojects.php)

Rules & Processes (/projects/dev_process/index-quick.php)

FAQ (/projects/dev_process/index.php)

Tools for committers (/projects/tools)

[Back to the top](#)

Save the date for OCX 2027

OCX 2027 will take place on 13-15 April 2027 in Brussels, Belgium.

[Learn more \(https://www.ocxconf.org/event/2027/summary\)](https://www.ocxconf.org/event/2027/summary)

Eclipse Foundation

[About \(http://www.eclipse.org/org/\)](http://www.eclipse.org/org/)

[Projects \(https://projects.eclipse.org/\)](https://projects.eclipse.org/)

[Collaborations \(http://www.eclipse.org/collaborations/\)](http://www.eclipse.org/collaborations/)

[Membership \(http://www.eclipse.org/membership/\)](http://www.eclipse.org/membership/)

[Sponsor \(http://www.eclipse.org/sponsor/\)](http://www.eclipse.org/sponsor/)

Legal

[Privacy Policy \(http://www.eclipse.org/legal/privacy.php\)](http://www.eclipse.org/legal/privacy.php)

Terms of Use (<http://www.eclipse.org/legal/termsofuse.php>)

Compliance (<http://www.eclipse.org/legal/compliance/>)

Code of Conduct (http://www.eclipse.org/org/documents/Community_Code_of_Conduct.php)

Legal Resources (<http://www.eclipse.org/legal/>)

Manage Cookies

More

Report a Vulnerability (<http://www.eclipse.org/security/>)

Service Status (<https://www.eclipsestatus.io/>)

Contact (<http://www.eclipse.org/org/foundation/contact.php>)





Support (<http://www.eclipse.org/projects/support/>)

Subscribe to our Newsletter

Enter your email address



**Follow
Us:**

 (<https://www.youtube.com/channel/UCej18QqbZDxuYxyERPgs2Fw>)  (<https://www.linkedin.com/company/eclipse-foundation/>)  (<https://www.facebook.com/eclipse.org/>)  (<https://twitter.com/EclipseFdn>)

Copyright © Eclipse Foundation AISBL. All Rights Reserved.