Eclipse Development Process

Contents

1. Purpose

This document describes the Development Process for the Eclipse Foundation. In particular, it describes how the Membership at Large, the Board of Directors, other constituents of the Ecosystem, and the Eclipse Management Organization (EMO) lead, influence, and collaborate with Eclipse Projects to achieve these Eclipse purposes:

> *The Eclipse technology is a vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools (the 'Eclipse Platform'). Eclipse Platform tools are exemplary in that they verify the utility of the Eclipse frameworks, illustrate the appropriate use of those frameworks, and support the development and maintenance of the Eclipse Platform itself; Eclipse Platform tools are extensible in that their functionality is accessible via documented programmatic interfaces. The purpose of Eclipse Foundation Inc., is to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.*

Explanatory comments, guidelines, and checklists - as well as additional requirements added by the EMO per section 3 - are noted in yellow boxes.

This document has the following sections:

- *Principles* outlines the basic principles upon which the development process is based.
- *Requirements* describes the requirements that the Eclipse community has for its development process.
- *Structure and Organization* specifies the structure and organization of the projects and project community at Eclipse.
- *Development Process* outlines the lifecycle and processes required of all Eclipse projects.

2. Principles

The following describes the guiding principles used in developing this Development Process.

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.
- **Meritocracy** - 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 tools, 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 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 platform quality frameworks capable of supporting the building of commercial grade products on top of them; and
- ship extensible, exemplary tools which help enable a broad community of users

2.3 Three Communities

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

- **Contributors** and **Committers** - 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 Committers and (non-Committer) Contributors. 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.
    - ♦ Projects must have diversity goals to ensure diversity of thought and avoid relying on any one company or organization. At the same time, we acknowledge that enforcing a particular diversity metric is a poor way to achieve these goals; rather we expect the project leadership to help the diversity evolve organically.
    - ♦ Diversity is a means to an end, not an end in itself, thus diversity goals will differ by project based on the other accomplishments of the project(s).
    - ♦ Projects are required to explain their diversity efforts and accomplishments during Reviews.
- **Users** - an active and engaged user community is proof-positive that the Project's exemplary tools are 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 organizations to participate. Like all good things, a user community takes time and effort to bring to fruition, but once established is typically self-sustaining.
- **Adopters** - an active and engaged adopter/plug-in developer community is the only way to prove that an Eclipse project is providing extensible frameworks and extensible tools accessible via documented APIs. Reuse of the frameworks 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 tools at the expense of extensible frameworks or vice versa.

2.4 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 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, tools, 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.

3. Requirements

This document and any additional criteria as established by the EMO contains requirements, recommendations, and suggestions.

R**Required** - Certain responsibilities and behaviors are required of participants in Eclipse open source projects. Projects that fail to perform the required behaviors will be terminated by the EMO. In keeping with the Guiding Principles, the number of requirements must be kept to an absolute minimum.

G**Guideline** - Other responsibilities and behaviors are recommended best practices. Collectively, we have learned that Projects are more likely to be successful if the team members and leaders follow these recommendations. Projects are strongly encouraged to follow these recommendations, but will not be penalized by this Process if they do not.

3.1 Requirements and Guidelines

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 Eclipse Project Development Guidelines to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products and services.

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.

4. Project Structure and Organization

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 when presented to the communities of developers, adopters, and users. Projects provide structure that helps developers expose their hard work to a broad audience of consumers.

Eclipse Projects are organized 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 **Sub-Project**. The term Project refers to either a Top-Level Project or a Sub-Project. Projects may be referred to as Sub-Projects or Components, but the choice of common name does not change the characteristics of the Project.

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
- IP Records
- Community Awareness

As defined by the Eclipse Bylaws - Article VII, the **Eclipse Management Organization (EMO)** consists of the Foundation staff and the Councils. The term **EMO(ED)**, when discussing an approval process, refers to the subset of the EMO consisting of the Executive Director and whomever he or she may delegate that specific approval authority to.

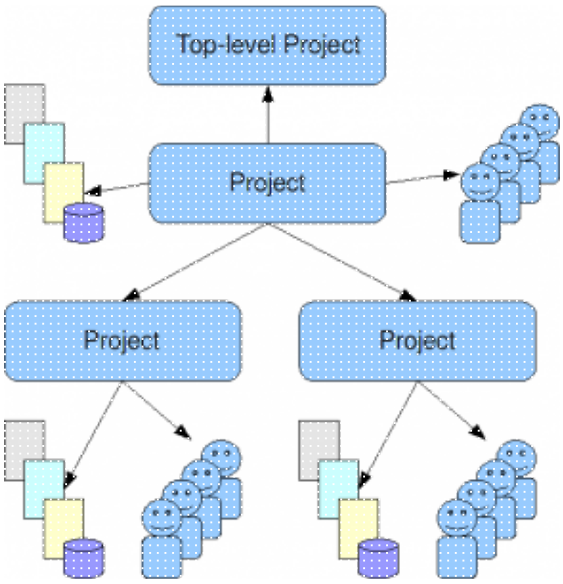
4.1 Committers

Each project has exactly one set of committers. Each Project's set of Committers is distinct from that of any other Project, including Sub-Projects 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 Sub-Projects.

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 sub-project does not give you any automatic rights on the "parent" project).

In practical terms, each Project has a single Unix group of its Committers that provides 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 Sub-Projects. Each of these Sub-Projects has its own distinct set of Committers and resources.



4.2 Code and Releases

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 unit, or container, for several Sub-Projects. Similarly, a Project might opt to provide a consolidated website, build and/or download site for its Sub-Projects (the Sub-Projects would then not require those resources for themselves).

Each Project has a single Bugzilla component for its bugs.

Any Project in the Mature Phase may make a **Release**. A Project in the Incubation Phase with two Mentors may make a pre-1.0 **Release**. A Release may include the code from any subset of the Project's descendants.

4.3 IP Records

A Project at any level may receive IP clearance for contributions and third-party libraries. IP approval will often include the same approval for all descendant Projects. However, IP clearance will only be granted at the most appropriate technical level.

4.4 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) or they may be part of a parent Project's communications (website, mailing list, forums/newsgroups, etc). 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, design discussions, and so on.

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

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 approves the Charter of each Top-Level Project.

Sub-Projects do not have separate Charters; Sub-Projects 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 Sub-Project 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 Project's top parent and by the EMO. A Project's Scope must be a subset of its parent's Scope.

4.6 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 Sub-Projects (if any) conform to the Eclipse Foundation IP Policy and Procedures.

The leadership for a Project is composed of the Project's Project Lead(s), the leadership of the parent Project (if any) and the PMC Leads and PMC Members for the Top-Level Project.

4.6.1 Project Management Committee (PMC)

Top-level projects are managed by a Project Management Committee (PMC). A PMC has one or more PMC 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 in particular, are ultimately responsible for ensuring that the Eclipse Development Process is understood and followed by their projects. The PMC is additionally responsible for maintaining the top-level project's charter.

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

4.6.2 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 Development Process, and that the project is engaging in the right sorts of activities to develop vibrant communities of users, adopters, and contributors. The initial project leadership is appointed and approved in the creation review. Subsequently, additional Project Leads must be elected by the project's Committers and approved by the Project's PMC and the EMO(ED).

In the unlikely event that a member of the Project leadership becomes disruptive to the process or ceases to contribute for an extended period, the member 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.

In exceptional situations, such as projects with zero active committers or projects with disruptive Committers and 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.

4.7 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.) and are expected to influence the Project's development.

See guidelines and checklists for electing a new committer.
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. 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. It is a responsibility that should be neither given nor taken lightly, nor is it a right based on employment by an Eclipse 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 of standard business days. 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), 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 Leaders 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 Project Leaders. (Unless otherwise specified, "an extended period" is defined as

"no activity for more than six months".)

Active participation in the user forums/newsgroups 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 forums/newsgroups.

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 and components. 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 the submitter. Committers are responsible for updating problem reports when they have done work related to the problem.

Committer, PMC Lead, Project Lead, and Council Representative(s) are roles; an individual may take on more than one of these roles simultaneously.

4.8 Councils

The Councils defined in Bylaws section VII are comprised of Strategic members and PMC representatives. The Councils help guide the Projects as follows:

- The **Planning Council** is responsible for establishing a coordinated Simultaneous Release (a.k.a, "the release train"). The Planning Council is further responsible for cross-project planning, architectural issues, user interface conflicts, and all other coordination and integration issues. The Planning Council discharges its responsibility via collaborative evaluation, prioritization, and compromise.
- See guidelines and checklists for the Architecture Council.
  The **Architecture Council** is responsible for (i) monitoring, guiding, and influencing the software architectures used by Projects, (ii) new project mentoring, and (iii) maintaining and revising the Eclipse Development Process Membership in the Architecture Council is per the Bylaws through Strategic Membership, PMCs, and by appointment. The Architecture Council will, at least annually, recommend to the EMO(ED), Eclipse Members who have sufficient experience, wisdom, and time to be appointed to the Architecture Council and serve as Mentors. Election as a Mentor is a highly visible confirmation of the Eclipse community's respect for the candidate's technical vision, good judgement, software development skills, past and future contributions to Eclipse. It is a role that should be neither given nor taken lightly. Appointed members of the Architecture Council are appointed to two year renewable terms.

4.9 Incubator Projects

A Project may designate a Sub-Project as an "Incubator". An Incubator is a Project that is intended to perpetually remain in the Incubation phase. 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.

Incubator Projects never have releases; they do not require yearly continuation reviews and they are not part of the annual release train. Incubators may have builds, and downloads. They conform to the standard incubation branding requirements and are subject to the IP due diligence rules outlined for incubating Projects. Incubators do not graduate.

The scope of an Incubator Project must fall within the scope of its parent project. The committer group of the 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). Incubator projects do not require Architecture Council mentors (the parent project's committers are responsible for ensuring that the Incubator project conform to the rules set forth by the Eclipse Development Process).

An Incubator project should designated as such by including the word "Incubator" in its name (e.g. "Eclipse 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 may create an Incubator. Incubators are created via a Creation Review. Alternatively, an Incubator can be created as part of a Graduation, Promotion, or Restructuring Review.

5. [Reserved]

6. 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.

All projects are required to report their status at least quarterly using the EMO defined status reporting procedures.

Projects must provide advanced notification of upcoming features and frameworks via their Project Plan.

6.1 Mentors

New Proposals that intend to do a Release are required to have at least two **Mentors**. New Proposals that will only Release code as part of a parent Project's Release are not required to have Mentors. Mentors must be members of the Architecture Council. The Mentors (including name, affiliation, and current Eclipse projects/roles) must be listed in the Proposal. Mentors are required to monitor and advise the new Project during its Incubation Phase, but are released from that duty once the Project graduates to the Mature Phase.

6.2 Project Lifecycle

Projects go through six distinct phases. The transitions from phase to phase are open and transparent public reviews.

6.2.1 Pre-proposal

See guidelines and checklists about writing a proposal.

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 by the EMO.

6.2.2 Proposal

See guidelines and checklists about gathering support for a proposal.
The proposers, in conjunction with the destination PMC and the community, collaborate in public to enhance, refine, and clarify the proposal. Mentors (if necessary) for the project must be identified during this phase.

- The Proposal phase ends with a Creation Review, or withdrawal.
- The Proposal may be withdrawn by the proposers.
- The EMO(ED) will withdraw a proposal that has been inactive for more than six months.

6.2.3 Incubation

See guidelines and checklists about incubation.
After the project has been created, the purpose of the incubation phase is to establish a fully-functioning 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.

- The Incubation phase may continue with a Continuation Review or a Release Review.
- Top-Level Projects cannot be incubated and can only be created from one or more existing Mature-phase Projects.
- The Incubation phase ends with a Graduation Review or a Termination Review.
- Designated Incubator Projects may remain perpetually in the Incubation phase; no reviews are required.

Many Eclipse Projects are proposed 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 the same time, however, the Incubation phase is useful for new Projects to learn the community-defined Eclipse-centric open source processes.

See guidelines and checklists for utilizing the Parallel IP process.
Only projects that are properly identified as being in the incubation phase (including designated Incubator Projects) may use the Parallel IP process to reduce IP clearance process for new contributions.

6.2.4 Mature

See guidelines and checklists about the mature phase.
The project team has demonstrated that they are an 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. Major releases continue to go through Release Reviews.

- Mature phase projects have Releases through Release Reviews.
- A Mature Project may be promoted to a Top-Level Project through a Promotion Review.
- A Mature Project that does not participate in a Release in given year may continue through a Continuation Review.

- Inactive Mature phase projects may be archived through a Termination Review.

6.2.5 Top-Level

See guidelines and checklists about being a top-level project.
Projects that have demonstrated the characteristics of a Top-Level Project (e.g., consistent leadership in a technical area and the recruitment of a wider developer community) can be promoted to Top-Level Project status. This promotion occurs through a Promotion Review. Upon the successful completion of a Promotion Review, the EMO(ED) may recommend that the project be promoted to the Board of Directors and ask that its Charter be reviewed and approved.

6.2.6 Archived

See guidelines and checklists for archiving projects.
Projects that become inactive, either through dwindling resources or by reaching their natural conclusion, are archived. Projects can reach their natural conclusion in a number of ways: for example, a project might become so popular that it is absorbed into one of the other major frameworks. Projects are moved to Archived status through a Termination Review.

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

6.3 Reviews

The Eclipse 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 of significant new technology or capability. It is a clear requirement of this document that members who are monitoring the appropriate media channels (e.g., mailing lists or RSS feeds) not be surprised by the post-facto actions of the Projects.

All Projects are required to participate in at least one Review per year.

For each **Review**, the project leadership makes a presentation to, and receives feedback from, the Eclipse membership.

A Review is a fairly comprehensive process. Gathering the material for a Review and preparing the presentation is a non-trivial effort, but the introspection offered by this exercise is useful for the Project and results are very useful for the entire Eclipse community. In addition, Reviews have a specific relationship to the requirements of the Eclipse IP Policy.

All Reviews have the same general process:

1. Projects are responsible for initiating the appropriate reviews. However, if a Project does not do so and the EMO believes a Review is necessary, the EMO may initiate a Review on the Project's behalf.
2. A Review then continues with the Project's Leadership requesting that the EMO(ED) schedule the Review.
3. Prior to the start of the review period, the Project leadership provides the EMO with review documentation.
    - The review documentation material always includes a document that describes the review.

The minimum contents of the document are specified by the individual Review types.
- The presentation material must be available in a format that anyone in the Eclipse membership can review. For example, Microsoft Powerpoint files are not an acceptable single format: such files may be one of the formats, but not the only format. Similarly for Apple Keynote files and Microsoft Word files. PDF and HTML are acceptable single formats.
- The presentation material must have a correct copyright statement and license.
- The presentation material must be *archival quality*. This means that the materials must be comprehensible and complete on their own without requiring explanation by a human presenter, reference to a wiki, or to other non-archived web pages.

4. The EMO announces the Review schedule and makes the documentation available to the membership-at-large.

The criteria for the successful completion of each type of Review will be documented in writing by the EMO in guidelines made available via the www.eclipse.org website. Such guidelines will include, but are not limited to the following:

1. Clear evidence that the project has vibrant committer, adopter and user communities as appropriate for the type of Review.
2. Reasonable diversity in its committer population as appropriate for the type of Review. Diversity status must be provided not only as number of people/companies, but also in terms of effort provided by those people/companies.
3. Documented completion of all required due diligence under the Eclipse IP Policy.
4. For Continuation, Graduation and Release Reviews, the project must have a current project plan, in the format specified by the EMO, available to the community.
5. Balanced progress in creating both frameworks and extensible, exemplary tools.
6. Showcase the project's quality through project-team chosen metrics and measures, e.g., coupling, cyclomatic complexity, test/code coverage, documentation of extensions points, etc.

The Review period is open for no less than one week and usually no more than two weeks of generally accepted business days.

1. The Review begins with the EMO's posting of the review materials at the start of the Review period
2. The proper functioning of the Eclipse Development Process is contingent on the active participation of the Eclipse Members and Committers, especially in Reviews, thus each Review has an EMO-designated discussion and feedback communication channel: a forum/newgroup, a mailing list, or some other public forum.
3. If a Committer election is required for a Review (for example, for a Creation Review), then it is held simultaneously with the Review period. Thus the election and the Review will end at the same time, allowing quick and efficient provisioning of the resulting Project.
4. The EMO(ED) approves or fails the Review based on the public comments, the scope of the Project, and the Purposes of the Eclipse Foundation as defined in the Bylaws.
5. The Review ends with the announcement of the results in the defined Review communication channel (the EMO(ED) will request that the Project Lead make this announcement).

If any Member believes that the EMO has acted incorrectly in approving or failing a Review may appeal to the Board to review the EMO's decision.

6.3.1 Creation Review

See guidelines and checklists about Creation Reviews.

The purpose of the Creation Review is to assess the community and membership response to the 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, as well as justify the initial Committers' participation in a meritocracy.

6.3.2 Graduation Review

See guidelines and checklists about Graduation Reviews.
The purpose of the Graduation Review is to confirm 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, and users
- operating fully in the open following the Principles and Purposes of Eclipse
- a credit to Eclipse and is functioning well within the larger Eclipse community

The Graduation Review is about the phase change from Incubation Phase to Mature Phase. If the Project and/or some of its code is simultaneously relocating to another Project, the Graduation Review will be combined with a Restructuring Review.

6.3.3 Release Review

See guidelines and checklists about Release Reviews.
The purposes of a Release Review are: to summarize the accomplishments of the release, to verify that the IP Policy has been followed and all approvals have been received, 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.

6.3.4 Promotion Review

The purpose of a Promotion Review is to determine if the Project has demonstrated the characteristics of a Top-Level Project, e.g., consistent leadership in a technical area and the recruitment of a wider developer community. The Project will already be a well-functioning Mature Eclipse Project, so evidence to the contrary will be a negative for promotion. Top-Level Projects, both through their existence and their Council memberships, have substantial influence over direction and operation of Eclipse, thus it behooves the membership to grant Top-Level status only for merit: for demonstrated service to the larger Eclipse ecosystem.

6.3.5 Continuation Review

The purpose of a Continuation Review is to verify that a Proposal or Project continues to be a viable effort and a credit to Eclipse. The Project team will be expected to explain the recent technical progress and to demonstrate sufficient adopter, developer, and user support for the Project. The goal of the Continuation Review is to avoid having inactive projects looking promising but never actually delivering extensible frameworks and exemplary tools to the ecosystem.

6.3.6 Termination Review

See Termination Review "How To" for more information.
The purpose of a Termination Review is to provide a final opportunity for the Committers and/or Eclipse membership to discuss the proposed withdrawal of a Proposal or archiving of a Project. The desired outcome is to find sufficient evidence of renewed interest and resources in keeping the Project or Proposal active.

6.3.7 Move Review

A Move Review is considered to be a special type of Restructuring Review.

6.3.8 Restructuring Review

The purpose of a Restructuring Review is to notify the community of your intent to make significant changes to one or more projects. "Significant changes" includes:

- Movement of significant chunks of functionality 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; and/or
- Change of project scope.

A Restructuring Review may include the movement of significant chunks of code. A move is considered significant if it has an impact on the community (i.e. if segments of the community will notice that the code has moved). This may include entire projects, bundles, and features, but likely excludes small fragments, code snippets and individual files. The IP Log of all moved code must be reviewed prior to the start of the review period (this, typically, is a subset of the project's IP Log). If all of the code is moved out of a project, a Termination Review for that project can be combined with the Restructuring Review.

Note that, regardless of whether or not a review is required, moving code from one Project to another is subject to the Eclipse IP Policy.

A Restructuring Review may necessitate the construction of one or more new projects. This tends to occur when an existing project is decomposed into two or more projects. In this case, a Restructuring Review is similar to a Creation Review. Any new projects that are created as part of a Restructuring Review must have their scope explicitly specified as part of the review. The scope of any new project must be a subset of the scope of the original project. Likewise, the set of committers assigned to a new project must be a subset of the committers of the original project (additional committers can be elected to the new project after it is created). Any new projects that fall outside of the scope of the original project, or wish to establish a different set of committers, must undergo the full project creation process.

Committers can be moved along with code into a new project as part of the project provisioning process. Committers cannot be moved along with code into an existing project. In this case, the existing project must elect the new committers into the project.

A project is expected to socialize pending changes using established communication channels prior to initiating the review. A Restructuring Review must provide the community with at least one week to review and respond to the changes. Prior to the start of that review period, the community must be provided with (via the EMO) completed review documentation that describes in specific terms what will be changed as part of the restructuring.

This may include:

- Name, description, scope, and committer lists of new projects that need to be created;
- Source and target locations for moves of source code directories;
- Reorganization of builds and downloads;
- Contribution questionnaires (CQs) that need to be moved or piggy-back CQs that must be created;
- Location of the approved IP Log; and
- Other information that helps the community understand the change.

6.3.9 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 Release Review involving itself and some or all of its child projects; a consolidated Restructuring Review may move the code for several projects; or a Release 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 Release and Graduation review, or aggregating a Release Review of a Project and several of it's child Projects generally makes sense. Combining Release Reviews for multiple unrelated projects most likely does not.

6.4 Releases

(Most of this section is borrowed and paraphrased from the excellent Apache Software Foundation Releases FAQ. 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.)

Releases are, by definition, anything that is distributed outside of the Committers of a Project. If users 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.

(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 are to be labeled `x.yMz`, e.g., 2.3M1 (milestone 1 towards version 2.3), 2.3M2 (milestone 2 towards version 2.3), etc.

- Release candidates are to be labeled `x.yRCz`, e.g., 2.3RC1 (release candidate 1 towards version 2.3).
- Official Releases are the only downloads allowed to be labeled with `x.y`, e.g., 0.5, 1.0, 2.3, etc.

All official Releases must have a successful Release Review before being made available for download.

(Exception 3: bug fix releases with no new features) Bug fix releases (x.y.z, e.g., 2.3.1) with no new features over the base release (e.g., 2.3) are allowed to be released without an additional Release Review. If a bug fix release contains new features, then the Project must have a full Release 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.

6.5 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. 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 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

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

8. Revisions

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

Due to the continued evolution of the Eclipse technology, the Eclipse community, and the software marketplace, it is expected that the Development Process (this document) will be reviewed and revised on at least an annual basis. The timeline for that review should be chosen so as to incorporate the lessons of the previous annual coordinate release and to be applied to the next annual coordinated release.

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 via an appropriate mechanism in a timely, effective manner.

8.1 Revision 2.6

This document was approved by the Eclipse Foundation Board of Directors in its meeting on XXXXXXXX. It takes effect (replacing all previous versions) on August 1/2011.