Eclipse Development Process

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.*

This document has five sections:

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

- *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.
- *Roadmap Process* describes the manner by which the EMO will work with the projects to create the annual Eclipse Roadmap.
- *Development Process* outlines the lifecycle and processes required of all Eclipse projects.

2. Principles

The following describe 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.
  - As a further explanation of Open, we include **Permeable** and **Receptive** - Projects are open to new ideas and new committers; not just in words, but in fact. In other words, those outside the core can, and do, influence and join the project.
- **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 Quality Culture

In this context, quality means extensible frameworks and exemplary tools developed in an open, inclusive, and predictable process involving the entire community.

- From the "consumption perspective," Eclipse Quality means good for users (exemplary tools - cool/compelling to

use, indicative of what is possible) and ready for plug-in developers (deliver usable frameworks with platform-ready APIs). The direct involvement of developers in use, reuse, testing, and feedback of components is essential to improving the quality.
- From the "creation perspective," Eclipse Quality means working with an open, transparent, and receptive process, open (and welcoming) to participation from technical leaders, regardless of affiliation.
- From the "community perspective," Eclipse Quality is that the community perceives quality, i.e., if the frameworks and tools are good enough to be used, then they have sufficient quality.
- From the "collective perspective," Eclipse Quality is the integration of the disparate Eclipse components into a unified whole, i.e., the value of the Eclipse brand as a whole over the individual projects.

The Eclipse projects are managed by different people and different organizations, most already experienced in software engineering. We want to ensure that we share the best practices of all our experts so that all projects benefit. We need to ensure that we have an "Eclipse committer community" rather than dozens of smaller "project committer communities".

2.3 Collective Reputation

Having the Eclipse name on a project provides a certain "goodness" to the project. And having great and amazing projects under the Eclipse banner provides a certain "goodness" to Eclipse. Correspondingly, having a highly-visible poor, closed, and/or failing project under the Eclipse banner detracts from that reputation.

A certain number of failures are expected in any research and development effort, thus we do not let the fear of failure prevent us from accepting interesting project. However, it is in the community's best interest to have a well-defined process for identifying and dealing with failures when they occur.

2.4 Eclipse Ecosystem

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;
- ship extensible, exemplary tools which help enable a broad community of users; and
- participate in the annual Roadmap process to ensure maximum transparency and bi-directional communication with the ecosystem.

2.5 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 the diversity goals to ensure diversity of thought and avoiding 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).

  - Project 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 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.

2.6 Clear and Concise

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.

2.7 Freedom, Autonomy and Evolution

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.

2.8 Just Enough Process

The Eclipse Development Process should be "just enough" to ensure that the community's goals (quality, openness, etc), but no more - we want to make it easy and inviting for high-quality teams to build interesting software at Eclipse. The entry bar should be "just high enough" to keep Eclipse great, but no more - we want to make it easy to experiment and explore new ideas while simultaneously supporting the ecosystem with strong releases. The entry bar should be "just high enough" to prevent Eclipse from growing too fast (because too rapid growth places too much of a strain on the mentoring capacity of the existing community) yet "just low enough" to prevent it from stagnating.

- The processes and goals should make projects:
  - Easy to propose
  - Fairly easy to create (e.g., enter incubation)
  - Kinda hard to graduate (e.g., exit incubation)
  - Pretty tough to ship
- The processes are designed to enhance the middle ground of continued quality growth in Eclipse projects by

eliminating the two undesirable endpoints:

- an entry bar so low that it results in a random collection of low-quality projects, and
- an entry bar so high that an insufficient number of new projects are created.

3. Requirements

This document and any additional criteria as established by the EMO contain 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. Structure and Organization

The Eclipse Projects are organized hierarchically. The top of the hierarchy are the set of **Top Level Projects**. Each Top Level Project contains zero or more **Projects** (for linguistic clarity, Projects as often referred to as **Sub-Projects**). Projects may contain one or more **Components**. Components are dependent, managed by the enclosing Project, and do not have independent release schedules.

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 Charter

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.

4.2 Scope

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 sub-projects and components 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 parent Top-Level Project and the EMO. A Project's Scope must be a subset of the parent Top-Level Project's Scope.

4.3 Management

Top-Level Projects are managed by a **Project Management Committee (PMC)**. Sub-Projects are managed by one or more **Project Leaders**. The Project Leaders, the Project Leaders of the Top-Level Project, and the PMC of the Project's Top-Level parent are referred to as the **Project Leadership Chain**.

PMC Leads are approved by the Board; PMC members and Project Leads are approved by the EMO(ED). The initial Project Leadership is appointed and approved in the Creation Review. Subsequently, additional Project Leadership (PMC members or Project Leaders) must be approved unanimously by the existing Project Leadership and the Board or EMO(ED) (as appropriate). 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 (a) if there are at least two other Project Leaders, then unanimous vote of the remaining Project Leadership; or (b) unanimous vote of the Project Leadership of the parent Project.

Each Project's leadership is required to:

- ensure that the Project is operating effectively by guiding the Project's overall direction and by removing obstacles, solving problems, and resolving conflicts
- ensure that all Project plans, technical documents and reports are publicly available and up-to-date
- notify the membership-at-large of all major new features and new code contributions via the defined membership notification mechanism (e.g., specific mailing list). This notification must occur in advance of the code being committed to the source repository.

See guidelines and checklists for notifying the membership of major new features.

- operate using open source rules of engagement: meritocracy, transparency, and open participation. These principles work together. Anyone can participate in a Project. This open interaction, from answering questions to reporting bugs to making code contributions to creating designs, enables everyone to recognize and utilize the contributions.
- work effectively with other Eclipse projects with which it has touchpoints.
-

Active participation in the user newsgroups and the appropriate developer mailing lists is a required responsibility of all Project Leaders and is critical to the success of the Project.

4.4 Project Team

Each Project has a **Development Team**, led by a Project Lead. 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 source code repository(ies) for the associated Project 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 using the process as specified in the Charter of the relevant Top-Level 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 has write access to the source code repository for the Project and/or website and/or bug tracking system. 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.

See guidelines and checklists for electing a new committer.

The nomination and election process for Committers to a Project must follow the Eclipse principles by being open and transparent.

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

4.5 Councils

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

- The **Requirements Council** is primarily responsible for the Eclipse Roadmap. There will always be more requirements than there are resources to satisfy them, thus the Requirements Council gathers, reviews, and categorizes all of these incoming requirements - from the entire Eclipse ecosystem - and proposes a coherent set of **Themes and Priorities**.
- The **Planning Council** is responsible for establishing a coordinated Simultaneous Release (a.k.a, "the release train") that supports the Themes and Priorities in the Roadmap. The Planning Council is 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.
- The **Architecture Council** is responsible for the development, articulation, and maintenance of the Eclipse Platform Architecture and ensuring the Principles of the Development Process through mentorship. 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 three year renewable terms.

See guidelines and checklists for the Architecture Council.

5. Roadmap Process

The Roadmap describes the collective Eclipse Projects future directions and consists of two parts:

1. **Themes and Priorities** from the Requirements Council
2. **Simultaneous Release Plan** from the Planning Council

The Roadmap must be consistent with the Purposes as described in Bylaws section 1.1. It is developed using the prescribed roadmap process.

The Roadmap is prepared by the Councils and approved by the Board annually. A proposed Roadmap or Roadmap update is disseminated to the Membership at Large for comment and feedback in advance of its adoption. This dissemination and all discussion and debate around the Roadmap must be held in an open and transparent public forum, such as mailing lists or newsgroups.

Prior to any Board vote to approve a Roadmap or Roadmap update, every Member has the right to communicate concerns and objections to the Board.

The process of producing or updating the Roadmap is expected to be iterative. An initial set of Themes and Priorities may be infeasible to implement in the desired timeframe; subsequent consideration may reveal new implementation alternatives or critical requirements that alter the team's perspective on priorities. The EMO orchestrates interaction among and within the Councils to drive the Roadmap to convergence.

This Development Process, the EMO, the Councils, and the Projects all acknowledge that the success of the Eclipse ecosystem is dependent on a balanced set of requirements and implementations. A Roadmap that provides too large a burden on the Projects will be rejected and ignored; similarly, a Roadmap that provides no predictable Project plans will be unhelpful to the business and technical plans being created by the ecosystem. A careful balance of demands and commitments is essential to the ongoing success of the Eclipse Projects, frameworks, and ecosystem.

The Project Leadership is expected to ensure that their Project Plans are consistent with the Roadmap, and that all plans, technical documents and reports are publicly available. To meet this requirement, each Project is expected to create a transparently available Project Plan that meets the following criteria:

1. Enumerates the areas of change in the frameworks and tools for each proposed Release
2. Consistent with and categorized in terms of the themes and priorities of the Roadmap
3. Identifies and accommodates cross-project dependencies
4. Addresses requirements critical to the Ecosystem and/or the Membership at Large
5. Advances the Project in functionality, quality, and performance

A Project may incrementally revise their Project Plan to deliver additional tasks provided :

1. the approved Roadmap is not put in jeopardy; and
2. the work is consistent with the Project Plan criteria (as described above)

6. Development Process

All Eclipse Projects, and hence all Project Proposals, must be consistent with the Purposes and the then-current Roadmap.

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.

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 and the appropriate public announcement communication channels.

6.1 Mentors

New Proposals are required to have at least two **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 requirement once the Project graduates to the Mature Phase.

The Mentors must attend the Creation and Graduation Reviews and the Mentors must vote +1 for those Reviews to be successful. If the Mentors do not attend or do not vote positively, the Creation Review cannot be successful regardless of other feedback or votes.

6.2 Project Lifecycle

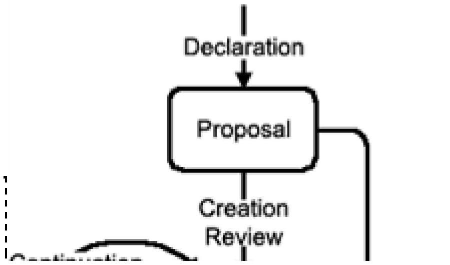
Projects go through six distinct phases. The transitions from phase to phase are open and transparent public reviews.

6.2.1 Pre-proposal

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.

See guidelines and checklists about writing a 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

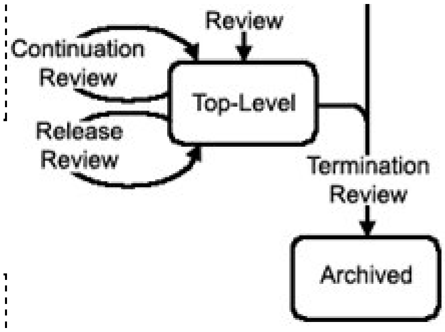
The proposers, in conjunction with the destination PMC and the community, collaborate in public to enhance, refine, and clarify

See guidelines and checklists

the proposal. Mentors for the project must be identified during this phase.

about gathering support for a proposal.

- The Proposal phase ends with a Creation Review or a Termination Review.



6.2.3 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 Top-Level Project.

See guidelines and checklists about incubation.

- The Incubation phase may continue with a Continuation Review .
- Top-Level Projects cannot be incubated and can only be created from existing Mature-phase Projects.
- The Incubation phase ends with a Graduation Review or a Termination Review.

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.

Only projects that are properly identified as being in the incubation phase may use the Parallel IP process to reduce IP clearance process for new contributions.

See guidelines and checklists for utilizing the Parallel IP process.

6.2.4 Mature

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.

See guidelines and checklists about the mature phase.

- Mature phase projects have Releases through a Release Review.
- A Mature Project may be promoted to a Top-Level Project through a Promotion Review.
- A Mature Project that does not 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

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.

See guidelines and checklists about being a top-level project.

6.2.6 Archived

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.

See guidelines and checklists for archiving projects.

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. It also ensures that the original or updated project goals are still consistent with the Purposes and Roadmap.

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 have 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. The Review process begins with the Project's Leadership requesting that the PMC approve the request for review.
    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. No less than one week in advance of the Review conference call, and preferably at least two weeks in advance, the Project leadership provides the EMO with the archival presentation material.
    1. The presentation material always includes a summary slide presentation or document. The minimum contents of the presentation are proscribed by the individual Review types.
    2. 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.
    3. The presentation material must have a correct copyright statement and be licensed under the EPL.
    4. 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.
4. The EMO announces the Review schedule and makes the presentation materials 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. Balanced progress in creating both frameworks and extensible, exemplary tools.
5. 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 itself:

1. Is open for no less than one week and no more than two weeks of generally accepted business days.
2. Begins with a conference call or other conference technology (e.g., web conferencing) so long as the technology is available to all members and incurs no additional costs to the attendees.
6. During the conference call, the Project Leadership (or EMO appointed Project representative) provides a brief summary of the reasons and justifications for the phase transition followed by a question and answer session.
7. After the conference call, the Review remains active for open and transparent discussion amongst the membership-at-large via the Project-appropriate mailing lists, newsgroups, or other designated forums.

8. At the end of the Review period, the EMO(ED) holds a public advisory vote of all Eclipse Members (which for clarity includes Committers Members as defined in the Bylaws 3.3(d)) using an announced voting mechanism.
    1. A successful advisory vote requires at least three +1s from the Membership external to the Project's Leadership Chain.
    2. A successful advisory vote requires at least one +1 from each level of the Project's Leadership Chain.
    3. A successful advisory vote requires no upheld -1s. An *Upheld -1* is a -1 that is followed within 24 hours by open, transparent, and public justification, and that justification is accepted by the EMO. Rejected -1s count as 0s.
    4. A successful advisory vote requires, if applicable, +1s from each of the Project's Mentors.
9. The EMO(ED) approves or fails the Review based on the public advisory vote, the scope of the Project, and the Purposes of the Eclipse Foundation as defined in the Bylaws.

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

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.

See guidelines and checklists about Creation Reviews.

The Creation Review must include a short bio of the proposed initial committers. Not a life story, but 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. (See [1])

6.3.2 Graduation Review

The purpose of the Graduation Review is to confirm that the Project is/has:

- a working and demonstratable code base of sufficiently high quality
- active and sufficiently diverse communities: 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

See guidelines and checklists about Graduation Reviews.

6.3.3 Release Review

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 Princples and Purposes of Eclipse.

See guidelines and checklists about Release Reviews.

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 existance 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 eco-system.

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

The purpose of a Termination Review is to provide a final opportunity for the Committers and/or Eclipse membership to discuss the proposed withdrawl 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.

6.4 Membership Involvement

The proper functioning of the Eclipse Development Process is contingent on the active participation of the Eclipse Members and Committers. The requirement for positive votes from Members (which includes Committers) which are not involved in the project will require each project to build relationships with other participants within the Eclipse community. The process is positive biased in that Reviews give more weight to negative votes (vetoes) as opposed to requiring a majority of positive votes to proceed.

Review vote summaries will include:

- Number and identification of all explicit positive (+1) votes
- Number and identification of all explicit abstentions (0)
- Number and identification of all explicit negative (-1) votes

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.
- **Inconsistent with Purposes.** It is alleged that a Project is inconsistent with the Roadmap and/or Purposes.
- **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. 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.

7.1 Revision 2.3.1

This document was approved by the Eclipse Foundation Board of Directors in its meeting on January 17, 2007. It replaces the original 2003 Development Process.