The Eclipse Development Process for Committers

eclipse.org/projects/dev_process

Copyright © 2012, 2018 The Eclipse Foundation. Made available under the terms of the EPL
Agenda

• Definition of an Eclipse Project
• The Eclipse Development Process
• Working as a Committer
• Releases and Reviews
• Project Management Interface
• Other Stuff You Should Know
What is an Eclipse Project?
TL;DR

An Eclipse Project...

• Conforms to the EDP and Eclipse IP Policy;
• Calls itself an “Eclipse Project” and conforms to Eclipse Foundation Branding Guidelines;
• Operates independently from any specific vendor;
• Uses infrastructure provided by the Eclipse Foundation for core content
Eclipse Development Process

Defines...

- The Open source rules of engagement
- Governance structure and definitions
- Relationships and roles

Provides a framework for project lifecycle management

- Releases and reviews
- Interaction between projects and the community
The Eclipse Intellectual Property Policy

The Eclipse IP Policy is concerned with establishing practices to mitigate legal risks associated with building products based on Eclipse software.

- Licensing
- Copyright
- Provenance tracking
Trademarks and Branding

The Eclipse Foundation holds all project trademarks on behalf of the community.

• Trademarks must be managed to remain valid
• Cycle of value
  • The Eclipse brand adds value to project trademarks
  • Project trademarks add value to the Eclipse brand
Source Code is Core Content

Source code must be *actively developed* on infrastructure provided by the Eclipse Foundation.

- Official project Git repositories must be the central authority
- Committers must push regularly to official project repositories
- Test code is source code

Level playing field; open to all comers.
Documentation is Core Content

Documentation is a first-class project resource.

- Documentation maintained using the same rules as source code
- Must be deployed on EF infrastructure
  - EF website, downloads, documentation server, ...
- May also be deployed elsewhere
  - e.g. Community website, ReadTheDocs
Issues/Bugs/Tickets are Core Content

Issues must be tracked using infrastructure provided by the Eclipse Foundation.

• Bugzilla
• GitHub Issues
Project Dev List is Core Content

The Project Dev list is the main means of contact between the Eclipse Foundation (staff and automatic services) and project team.

• Formal EF process must happen on the devlist
  • practical: votes, etc.
• Other project communication must be captured in a public channel
Non-Core Content

Other content must be transparent and open but not necessarily owned/operated by the EF

• User forum discussions, presentations
• Builds and downloads
• Examples, blogs, social media handles, …
“The Keys” Must be Kept Safe

Core infrastructure must be *immune* from domination by any single individual or organization.

• The EF must hold the keys for core infrastructure
• Pick up and carry on

Ownership of non-core infrastructure must be shared

• Must be able to survive departures of team members
• Open source rules of engagement
• Governance, structure, definitions, reviews
• General framework for projects
• Day-by-day development rules/process is defined by the project
Open Source Rules of Engagement

● Transparent
  - Project discussions, minutes, deliberations, project plans, plans for new features, and other artifacts are open, public, and easily accessible

● Open
  - The same opportunity to all
  - Everyone participates with the same rules

● Meritocracy
  - The more you contribute the more responsibility you will earn
  - Committer/Project lead elections
Vendor Neutrality

● Open to all comers, including competitors!
● Vendor neutral resources and services
  - Project website
  - Issue tracker
  - Mailing lists and forums
  - Downloads
  - VCS/Git
● Committers must work in the open
● Freedom of action
  - GitHub services backed up on EF servers
Three Communities

● Users
  - Users are, well... users

● Adopters
  - Individuals, groups, organizations
  - Build products, extensions, based on your project

● Developers
  - Contributors, committers
Writing code is fun, but...

- Open source rules of engagement
  - Transparency, openness, meritocracy
- Have project-specific diversity goals
  - Building diversity takes work
- Actively court contributors
- Be responsive when they do come
  - “Kill with kindness”
Project Leadership

● Project lead(s)
  - Management, technical leadership
  - Primary liaison with PMC and EMO

● Project Management Committee (PMC)
  - Technical approval of IP contributions
  - Approval of releases
  - Project oversight, general responsibility

● EMO
  - Overall responsibility to manage processes and maintain services
  - Eclipse Foundation Staff, Architecture and Planning Councils
Working as a Committer
Committers and Contributors

• Committers
  • Have write access to project resources (Git repos, downloads, etc.)
  • Must complete committer paperwork
    • Individual or Member Committer Agreement (ICA/MCA)

• Contributors
  • Contributions must be accepted by a committer
  • Must electronically sign the Eclipse Contributor Agreement (ECA)
  • May be invited to join the project as a committer
Eclipse Contributor Agreement

- Contributors must digitally sign the ECA
  - Confirms that contributor agrees to comply with the DCO
- Developer’s Certificate of Origin
  - You have authored 100% of the content
  - You have the necessary rights
  - Provided under the project license(s)
  - Your contributions are public
- Contributors must “sign-off” on each commit

https://eclipse.org/legal/ECA.php
Contributions via Git

commit d6cf5241f1377a039fc2906378711091a26e932cb
Author: Some Body <somebody@somewhere.com>
Date: Wed May 29 16:17:36 2013 +0200

[123456] Hide unwanted common navigator action bar items

This change hides unwanted 'Link with Editor' and 'Customize View...' items from the local toolbar and the view menu.

See bug 409722 for restoring the feature of activating the editor when changing the selection using 'Link with Selection'.

Bug: https://bugs.eclipse.org/123456
Change-ID: Ia2bd5091303d1b0a738157effc24e4dac5a7d0c7
Also-by: Some Bodyelse <somebodyelse@nowhere.com>
Signed-off-by: Some Body <somebody@somewhere.com>
Handling Git Contributions

- Git/Gerrit will check for ECA/sign-off
- GitHub will warn if ECA or sign-off is missing
- Engage with IP Team if required, e.g.
  - Contribution has >1 KLOC
  - You’re not sure of the provenance
  - Not certain that the ECA/DCO terms are met
Intellectual Property
Due Diligence Process

- All code in project VCS or downloads
  - All third-party libraries must be vetted by the IP Team
  - Initial contribution
  - Many contributions must be vetted as well, e.g. >1KLOC

- Vetting
  - Known pedigree
  - Permissive license; compatible with project license (e.g. EPL)

- Due diligence process defined
Dependencies
Prerequisites and “Works With”

Diagram showing dependencies between Eclipse Project Content, consumer, third party content, and "Works With" content. The diagram illustrates the flow of requirements and optional dependencies, highlighting the need for vetting by the IP Team.

Copyright (c) 2012, 2018, Eclipse Foundation, Inc. | Made available under the Eclipse Public License 2.0 (EPL-2.0)
Contribution Guide

- Git, Gerrit, GitHub, ...
- Issue tracking (Bugzilla)
  - “Help wanted” issues
- Project plan
- How to build
- How to engage (mailing lists, forums, issues)
- Spend time with the community
Keep Project Information Up-to-date

- Project and release metadata, website, downloads
- Project code must be buildable
- Maintain a contribution guide
- Keep project communities informed
Releases and Reviews
Releases
Release Review

● Major/minor releases
  - Release review
  - IP Log approval
  - Plan to spend time planning/documenting release

● Service/Bugfix-only releases
  - No review
  - No IP Log approval
Release Review
Project Management Interface (PMI)
Project Metadata

● Used for discovery and drives various processes and systems
  - e.g. IP Log generator, notices to members and committers
● Description, scope, logo, technology type
  - “Elevator pitch”
● Source code, builds, issue tracker
● Releases, relationships to other projects, build technologies
Other Stuff You Should Know
IP Terms

- Initial Contribution
- Contribution Questionnaires (CQs)
- Piggyback CQ
- IPZilla
- Parallel IP
IT Services

- Connect with the Webmaster for IT support
  - Git repositories, builds, downloads
  - Permissions
- Connect via Bugzilla (Community)
- webmaster@eclipse.org
Eclipse Project Handbook

- All-in-one guide for project leads and committers
- Starting a new project
- Forge-specific variations:
  - https://www.eclipse.org/projects/handbook
  - https://www.polarsys.org/documentation/handbook
Incubation Mailing List

- Connect with other incubating projects
- Connect with mentors
- http://dev.eclipse.org/mailman/listinfo/incubation
- incubation@eclipse.org
Things that We Didn't Talk About

- Committer and Project Lead Elections
- Graduation from Incubation Phase
- Committer Paperwork
- Starting an open source project
- IP Logs
Links

- Eclipse Development Process
  - http://www.eclipse.org/projects/dev_process
- Eclipse Project Handbook
The End