

The Eclipse Development Process

Wayne Beaton
The Eclipse Foundation

Eclipse Development Process

Rules of Engagement

Rules of Engagement

- Transparency
 - Invite participation
- Openness
 - Accept participation
- Meritocracy
 - Earn participation

Transparency

- Project discussions, minutes, deliberations, project plans, plans for new features, and other artifacts are open, public, and easily accessible.
 - Bugzilla
 - Mailing lists, Forums/Newsgroups
 - Code

Open

- The same opportunity to all
- Everyone participates with the same rules
- There are no rules to exclude any potential contributors
 - Including direct competitors in the marketplace

Meritocracy

- The more you contribute the more responsibility you will earn
- Development and leadership roles in Eclipse are merit-based and earned by peer acclaim

The Eclipse Development Process

The Four Cs

The Four Cs

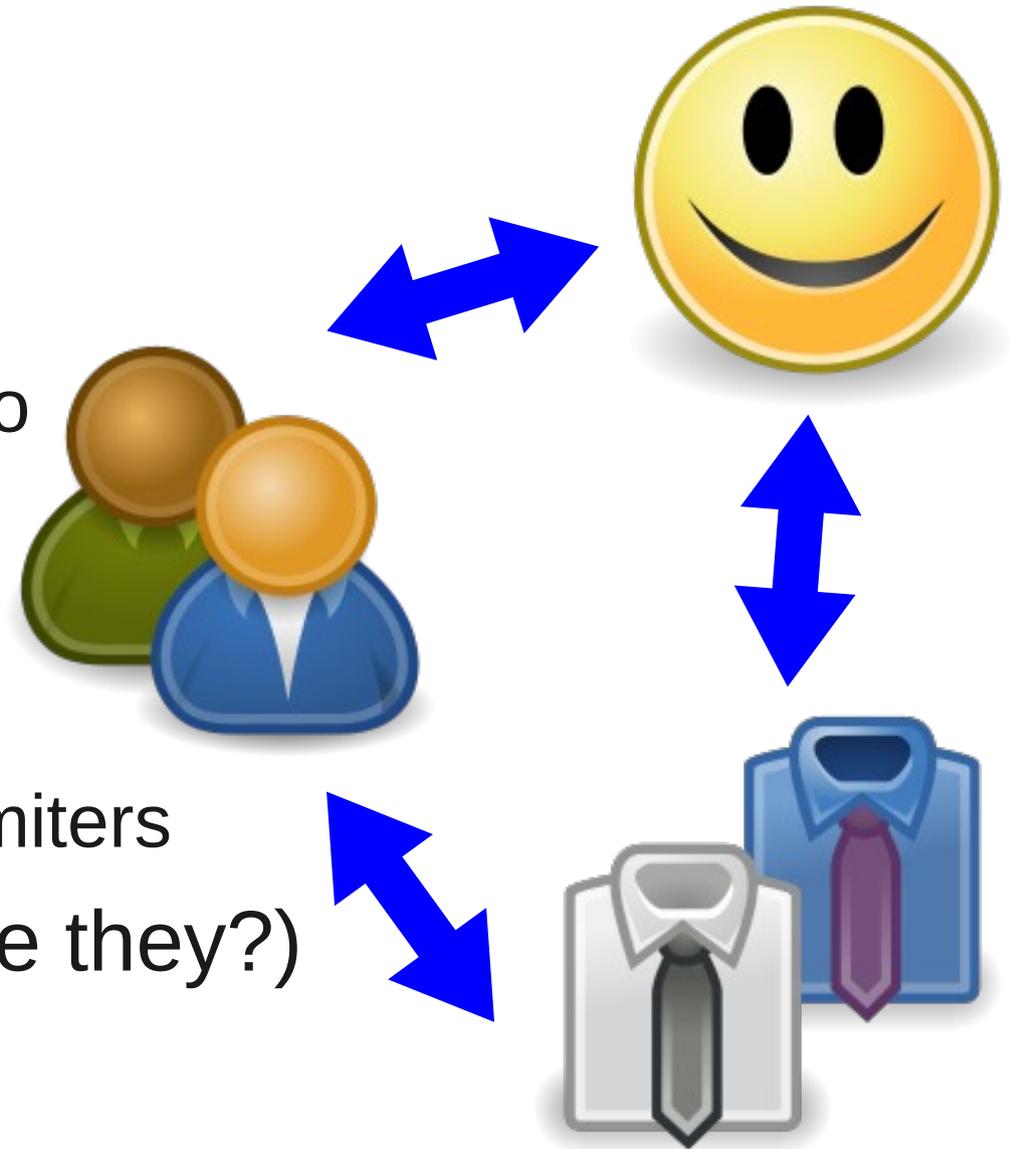
- Code
- Community
- Cleanliness
- Cwality

Code

- Code is... well... code.

Community

- End Users
- Adopters
 - Groups, individuals who base products on your project's code
- Developers
 - Contributors and Committers
- All are important (or are they?)



Cleanliness

- Where does the code come from?
 - Who owns the copyright?
 - Is the owner really the owner?
 - What license does the owner grant?
- All code is subject to the Eclipse Due Diligence Process

Cwality (Quality)

- Transparent issue tracking, dev list discussion
- Reviews
- Inviting and accepting participation
- Diversity

The Eclipse Development Process

Who's who?

Eclipse People

- Eclipse Management Organization (EMO)
 - Eclipse Foundation Staff
 - Planning Council
 - Architecture Council
- Project Management Committee (PMC)
- Project Lead
- Committer
- Contributor

The Eclipse Development Process

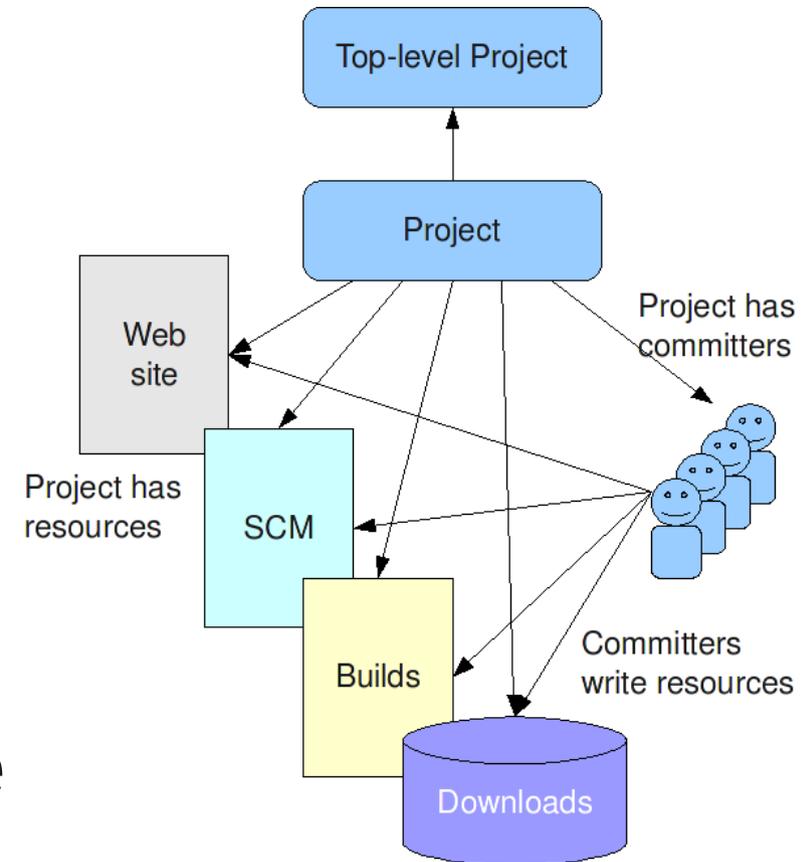
Project Structure

Project

- The means by which open source work is organized
 - Project Leaders, committers
 - Code, releases, builds, downloads, websites, and more
 - IP Records
 - Community Awareness
- All open source software development occurs within the context of a Project

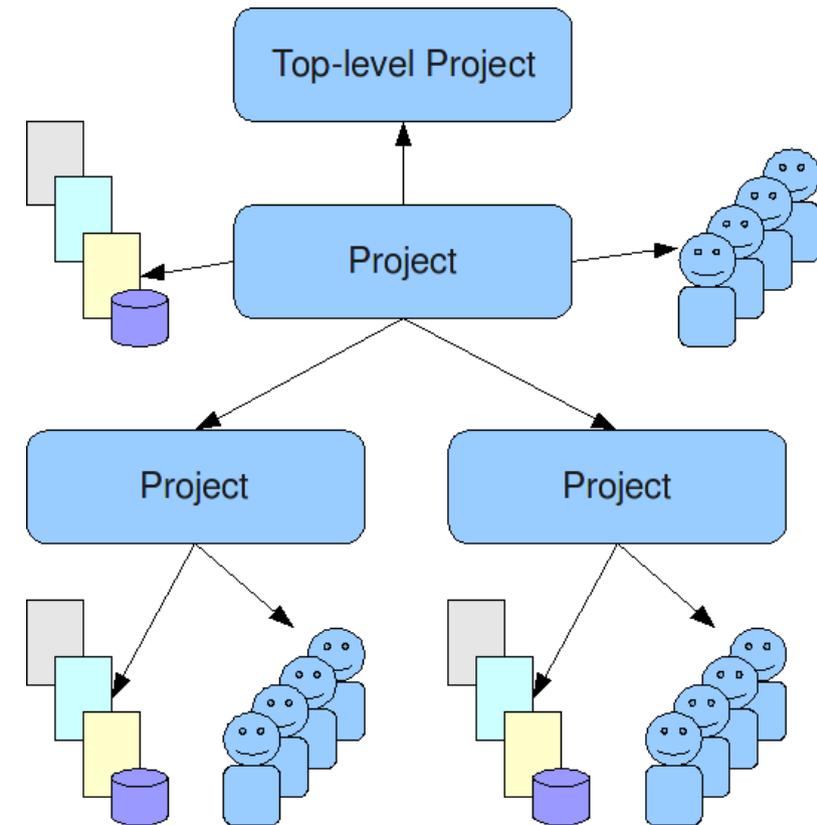
Project Structure

- Optionally have:
 - SCM (Git)
 - Website
 - Time on build server
 - Space on download/archive server
- Each project has exactly one UNIX Group/set of committers



Subprojects

- A project may have zero or more subprojects
- Each subproject has its own committers and resources
- Subproject scope must align with parent project scope
- Parent project leadership provides oversight



Lifecycle

- Stages
 - Proposal
 - Incubation conforming/non-conforming
 - Regular
 - Archived
- Projects undergo regular public reviews

Permanent Incubators

- Subproject of a mature project
 - Has "incubator" in the name
- No reviews, No releases, No plan
- Purpose:
 - Experiment with new ideas
 - Benefit from Parallel IP
 - Grow new committers
- Easy to create

The Eclipse Development Process

Starting a Project

Starting a Project

- Proposal assembled, delivered to EMO
- EMO and PMC review/feedback
- Posted for public review
 - Minimum of two weeks
- Creation review
- Project is provisioned
- Submit your initial contribution to the IP team
 - Wait for approval

Proposal Document

- Project name, top-level, parent, ...
- Description, Scope, Background
- Why Eclipse?
- Initial Committers, Project Lead, Mentors
- Initial Contribution
- Legal issues
- Interested Parties
- Schedule

The Eclipse Development Process

Reviews and Releases

Schedules and Releases

- Projects set their own release schedule
- Release plan created at beginning of a cycle
 - Revised throughout the cycle
- Review documentation at the end of a cycle
 - Retrospective for the committers
 - Preparation for adopters
 - Notification for users
- Every release requires a community review

Project Plan

- Required for every release
 - Introduction
 - Release Deliverables
 - Release Milestones
 - Target Environments
 - Compatibility with Previous Releases
 - Themes and Priorities
- Modified throughout development cycle

Review documentation

- Features, APIs, Standards, Non-Code Aspects
- Architectural, Security Issues
- Usability
- End-of-Life
- Bugzilla
- Schedule
- Communities
- IP Issues

IP Log Review

- IP Log Tracks:
 - Licenses;
 - Past and present committers;
 - Third-party libraries; and
 - Contributions from outside the project (i.e. non-committers)
- IP Log must be approved by the IP Team prior to each release
- Automatic IP Log generator does the work!

“Service” Releases

- “Bug fix” release
 - No API changes
 - No new functionality
- Third-segment change
 - e.g. 3.7.0 to 3.7.1/3.7.2
- No release review is required for service releases

Other Reviews

- Restructuring Review
- Graduation Review
- ~~Continuation Review~~
- ~~Promotion Review~~
- Termination Review

The Eclipse Development Process

Be successful

Define Success

- Define “success” realistically
- How large is your potential community?
- Is your technology niche or mainstream?
- An academic research project probably won't develop a community of millions
 - But that's okay. Just be realistic about the potential size of your community

Transparency and Openness

- Transparency: let everybody see what you're doing
- Openness: let everybody participate
- Project plans must be made in public
 - With input from the community
- Let other contribute
- Be open to new ideas, different approaches

Diversity

- Users and adopters see diversity as a way of protecting their investment in your technology
- The project should outlive your involvement
 - Get multiple companies/organizations involved
 - Get competitors to participate
- Generalize to balance competing needs and goals

Be Proactive

- Find community
 - Don't assume that community will find you
- Speak at conferences, user groups, technical briefings, ...
- Be prepared to visit prospective adopters and work directly with them
- T-shirts get you part of the way, but to really be successful, you have to provide support
 - At least initially

“Grass Roots”

- Build an dedicated group of believers
 - They'll do most of the work for you!
- Support your followers
 - Be a force multiplier

Make Community a Priority

- Schedule regular time to interact with each of your communities
- Do the Little Things:
 - Bug reports
 - Questions
 - Feedback
 - Respond in a timely manner
- Set aside time every day

Access

- Make it easy to find your stuff
- Provide downloads
- Keep web page up-to-date
- Solicit linkage from related sites

The Eclipse Development Process

Wrap Up

Links

- The Eclipse Development Process
 - http://www.eclipse.org/projects/dev_process/development_process.php
- Development Resources
 - http://wiki.eclipse.org/Development_Resources

Gene Simmons Never Had Eclipse When He Was a Kid



Photo by Rodney Gitzel, from Drop-D Magazine
<http://dropd.com/issue/23/KISS/>

How do we know? We know because our own well-documented research has shown conclusively that a child who lacks a personal computer running Eclipse during those earliest school years will very probably grow up to be a bass player in a heavy-metal rock band who wears women's fishnet pantyhose and sticks his tongue down to his kneecaps. Just like Gene Simmons.

Your child's future doesn't have to look like this.

Eclipse, complete with Java™, web, C/C++, and PHP development; modeling; and data, reporting, test, and performance tools is just what your four-year-old needs to compete in today's cut-throat world of high expectations.

Eclipse... Get it before it's too late. Gene's mother wishes she had.

