Running a Successful Open Source Project

Wayne Beaton,
Gunnar Wagenknecht
Who we are and why we are here!

Wayne Beaton
@waynebeaton
Director of Open Source Projects
Eclipse Foundation

Gunnar Wagenknecht
@guw
Principal Member Technical Staff,
Salesforce
Define “Successful”
Understand Open Source
(at least a little)
A Typical OSS Project

Is there such a thing?

Transparency
Openness
Meritocracy
Vendor Neutrality
Code and Documentation
Rules

Copyright © 2016, 2017 The Eclipse Foundation and Salesforce.
Made available under the CC-BY-3.0
People!

Community Roles

Owners
Leaders
Developers/Committers
Contributors
Community Members
... and more
The values of communities

Or why do we have them?

Outreach
Brand
Culture
(Moral) Support
...

Copyright © 2016, 2017 The Eclipse Foundation and Salesforce.
Made available under the CC-BY-3.0
Open Source Foundations

Why move your project to a Foundation?

Vendor neutrality
Governance model
Resources and Services
Marketing
...

Copyright © 2016, 2017 The Eclipse Foundation and Salesforce.
Made available under the CC-BY-3.0
Build Your Community
Facilitate Success

- Establish participation rules
- Operate transparently
- Be open to new ideas
- Make building easy
- Make reporting issues easy; respond
- Lower barriers

Stop asking “where’s the patch” with a snarky tone.
Listen

Don't use Twitter to report bugs. Just don't.
Conferences
Participate in discussions
Contribute Code
Code

… is about more than code

Bug fixes, new functionality
Coding conventions
License
Intellectual property, copyright
3rd party libraries

…
Select your license…

… carefully.
Manage Copyright

- Who owns the code?
  - The owner may be the author’s employer (check your contract)
  - Do you assign ownership when you contribute?
- The project itself is likely not a legal entity
- State the license (use SPDX code)

/*================================================================*****
* Copyright (c) 2017 The Eclipse Foundation, and others.
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the Eclipse Public License v1.0
* which accompanies this distribution, and is available at
* 
* SPDX-License-Identifier: EPL-1.0
* 
* Contributors:
* Wayne Beaton - initial API and implementation
/*================================================================*****
Establish Guides and Conventions

- Put a Contribution guide in your repository
- Coding conventions
- Source code formatting
- Patch size
- Unit tests
- Code comments
- Separate commits/changes (avoid fixing B while working on A)
- Communication & expectations
Track Intellectual Property

- How is the project code licensed?
- Who owns the code you wrote?
- Are you allowed to contribute?
- Are you allowed to (re-) license the code you are submitting?
- Use a Contributor (License) Agreement (CLA/CA)?
- Define an intellectual property management process
- What about third party content?
Intellectual Property Due Diligence is Hard Work

- License
  - Are the licenses in the contribution compatible with the project license?
- Provenance
  - Did the people who claim to have authored the code actually author the code?
- Integrity
  - Are the license statements valid?
  - Has the license changed?
  - Has code been inappropriately copied?
Require a Contributor Agreement/Certificate of Origin

Generally some combination of:

- The contributor wrote the code
- Contributor has necessary rights to submit the code
- Provided under the terms of the project license
- A public record of the contribution is maintained indefinitely
Protect Your Trademark

- Who owns the project/product name?
- Should you register a trademark?
- Foundations hold the name on behalf of the community
  - Prevent any single vendor/individual from dominating the project
- Trademark usage guidelines
- Leverage the brand/grow the value of the brand
What we Didn’t Talk About

- Testing
- Support (end user vs. adopter vs. internal)
- Business drivers for getting involved
- Diversity and longevity
Why should you care about all of this?
Thank you!