

# Open Source Development Business Practices

Richard Fontana



# Agenda

- Education, policies and processes
- Advising developers on open source

# Education, policies and processes

# Education

- Employee decisionmakers need adequate IP training
  - Open source is just one facet of IP management
  - Varies with employee role (e.g. engineer vs. sales)
  - Revise as business evolves, lessons are learned
- Legal departments need open source expertise
- Hire seasoned FOSS developer employees



# Open source policies?

- What's so unique about open source?
- Need comprehensive software development policies
  - Use of third-party code
  - Desirable product licensing models
  - Ensure legal compliance in development and distribution
- Need stakeholder buy-in
- Reflect existing corporate practices/requirements

# Open source *processes*

- Decide how decisionmaking is delegated
- Have clear "owners" of issue categories
- Harmonize procedures with corporate culture
- Avoid over-bureaucratization and over-lawyering
- Avoid wheel-reinvention, but review safe harbors



# Basic goal: avoiding infringement

- Containment of real (typically small) legal risk
- Building and securing good community relationships
- Depends on resources, risk tolerance, circumstances
- Lower risk  $\Rightarrow$  higher tolerance of indeterminacy

# Some risk factors

- Nature of code use (distributed, internal only, cloud?)
- Core product code vs. community source code
- Replaceability/removability of code
- Predictable dispute posture: lawsuit vs. amicable fix
- Predictable consequences of (an unlikely) lawsuit



# Using "important" community project code

- Demands care beyond mere legal risk containment
- Licenses often only approximate cultural expectations
  - License interpretation tied to community tradition
  - Companies should try to understand and meet expectations
  - Engage with projects ⇒ shape legal expectations

# Advising developers on open source

# Situations

- Using upstream open source code
- Participating in upstream projects
- Initiating new projects



# Using upstream code

# Using upstream code

- Two cases
  - Incorporation of isolated source code
  - Use of whole libraries/dependencies
- Developers need to understand:
  - Mere public availability  $\neq$  acceptability
  - Legal terms (or their absence) matter

# Upstream code: common issues

- Ascertain legal terms governing third-party code
  - Know to get help from counsel in unclear cases
  - Contact upstream developers to clarify or negotiate
- Avoid propagating code with missing legal information
  - Even if license is confirmed and acceptable
- Push mods upstream to reduce maintenance burdens



# Importing/bundling binary dependencies

- Greater uncertainties/complexities regarding license compliance and code provenance
- Encourage building from source
  - Source often contains legal information absent from binaries
  - Facilitates compliance with copyleft licenses

# Participating in upstream projects

# Benefits of participation

- Engineering
  - Maintenance (bugs, bug fixes, feature requests, enhancements)
  - Influencing technical direction and priorities
- Legal
  - Minimizes risk of legal conflicts with upstream
  - Arm's length relationship may have legal benefits
  - Influencing legal decisionmaking



# "Off-hours" participation

- Possibility of conflict between company and employee
  - Differing expectations regarding IP ownership
  - True conflicts of interest sometimes arise
  - Address issue to resolve doubts early on
  - Have clear guidelines and approval policies
- Consider liberalizing policies
  - Increased influence and expertise of your developers
  - Increased job satisfaction; talent recruitment

# Scrutinize project governance

- Who is in control of the project?
- True community projects vs. corporate projects
- Contributor agreements

# Initiating open source projects



# Initiating open source projects

- Motivations (adoption, R&D/development methodology, attract customers, employee interest)
- Two categories
  - Previously proprietary/internal code
  - Starting from scratch as open source
- Legal scrutiny should be proportionate to business significance and substantiality

# Some legal issues

- License choice and third-party license compliance
  - License consistent with business goals and policies
  - License consistent with third-party terms
  - Use popular, standard licenses where possible
  - Understand tradeoffs represented by different license policies
- Patent issues (mining; portfolio impacts)
- Trademark issues (clearance; registration; TM policies)



# Do:

- follow contemporary project development norms
- have transparent governance
- be prepared to give up total control
- consider participation/affiliation with other organizations
- use a public, web-browsable, versioned source repo

# Don't:

- use unconventional open source licenses
- use pseudo-open-source licenses
- leave legal terms unclear
- throw source code over the wall
- have policies that discourage community-building



# Thank you!

[gplus.to/fontana](https://plus.to/fontana)

[@richardfontana](https://twitter.com/richardfontana)

[aleatoric.org](https://aleatoric.org)

[rfontana@redhat.com](mailto:rfontana@redhat.com)

