Introduction to Open Source

Carl Eric Mols
Mateo Davis

Sony Mobile Communications AB

Agenda

• General Overview
  • What is it?
  • Theory of Why it Works
  • Open Source Licensing Basics

• Open Source at Sony Mobile
  • History
  • Strategy
  • Intellectual Property
  • Open Source Processes
Why are we here??

- SOMC’s History/Transition – Share lessons learned
- Engineers/Developers - have tremendous opportunities, but with that comes responsibility
- Lawyers and Engineers need to work together – legal and technical issues overlap

What is Open Source Software?

- Computer Software that provides its source code under a license with the rights to study, change, and distribute the software to anyone and for any purpose
- OSI Definition: 10 Freedoms – http://opensource.org/osd-annotated
- Is it widely used?
Wildly Successful

- More Stable
- Faster Development Time
- More Secure
- It is everywhere:
  - 2 billion people own a smartphone, full of Open Source
  - Internet, web services, medical equipment, cars and likely your fridge too.
  - Even the US DoD is a major user of Open Source … think ARPAnet …

The Cathedral vs The Bazaar – Why it Works

- Linus’ Law: *with enough eyeballs, all bugs are shallow*
- Release early, release often (bleeding edge vs. stable releases)
- Users = Co-Developers: users become testers, and testers become contributors
- Skilled developers wandering through your architecture
- Self-selection of developers
- Public - your work is on display (and the criticism can be brutal)
- Meritocracy + Democracy/little management
- Crowd Power – crowd decision-making works in the context of code development
- Why developers volunteer: reputation, experience, scratch an itch/solve a problem
- Why developers are paid: shared support and maintenance makes it cost efficient and more reliable

**Licensing Basics - Copyright**

- Copyright attaches to every novel expression of an idea (e.g., software)
- Exclusive Right to:
  - Copy
  - Modify
  - Distribute
- Time Limited – 50+ years
- License – grants defined rights

**Copyleft**

**Richard M Stallman**

Reciprocity

Distribution = trigger

Changing the code/combining with other code = need to provide changes/combination to the downstream recipient
Open Source Licensing Overview

CopyLeft

Permissive

<table>
<thead>
<tr>
<th>License</th>
<th>Uptake</th>
<th>Easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPLv2</td>
<td>Force Feedback/limits usage</td>
<td></td>
</tr>
<tr>
<td>(+ v3)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(+ Affero)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mozilla Public</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apache 2.0</td>
<td>Uptake</td>
<td>Easy</td>
</tr>
<tr>
<td>MIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notice work only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patent protection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modifications to files</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

License Interpretation (Re Copyleft)

- GPL uses very technical language (+ is a manifesto as much as a license)
- Glosses on licenses created by old message board discussions/FAQs
  - E.g., Linus Torvald’s interpretation of the GPL as it applies to Linux
  - Stallman/FSF’s FAQ pages about the GPL
- What is a derivative work often becomes very technical
- Widely recognized exceptions/interpretations by the community – shims, sockets, etc.
Open Source Community

- Free Software Foundation – Stallman founded
- Software Freedom Law Center – Eben Moglen
- Apache Foundation
- Linux Foundation
- gpl-violations.org

Open Source at Sony Mobile
Overview of Different Open Source Activities

- **TOOLS** (Gerrit, etc)
- **ANDROID**
  - Applications
  - Upstream Components (Webkit, Chromium)
- **Linux**
- **Firefox**
- + Miscellaneous
  - Dev. World Sample Code, Engineers’ side projects, Open Device Project, etc
  - IS/IT (Ubuntu, Apache, etc)

OS History

- 2002
  - Entry
  - OSE
  - Symbian
  - Microsoft
- 2009
- Present
  - Android
  - Firefox
  - Others ??
Official Strategy

Sony Mobile’s Strategy in Open Source is to reduce our OPEX and TTM through active but selective contributions of Non-Differentiating SW to the Open Source Community.

Fundamental objectives for [AOSP] contributions

• Reduce Cost of Maintenance
  Contributions reduce the patch mountain.

• Improve Time-To-Market
  Fewer patches means faster bring ups.
  Feedback from Google shows insights to roadmaps/strategic direction.

• Disrupt Competitors
  By contributing, we force competitors to re-adjust their set of patches.

• Get Closer to Google
  Create engineer-to-engineer relationships that can be leveraged to drive the business.
  Influence Android strategy.
**Contribution Strategy chart**

- **Business Impact / Uniqueness**
  - High
  - Low

- **Control Complexity**
  - High
  - Low

- **Platform / Leverage**
- **Cost focus**
- **TTM focus**
- **Product / Bottleneck**
- **Strategic Alliances & Investments**
- **Control focus**

---

**Major groupings**

- **Commodity**
- **Opportunity**
- **Services and Apps**
- **Enabling technologies**
- **Operator requirements**
- **Differentiating**
  - = Keep close, but enablers should be contributed.
- **Non-differentiating**
  - = Contribute
Intellectual Property Rights and Contributions

- **Patent Strategy**
  To reduce royalty payments and to ensure that products are the most appealing.

- **IPR Dept**
  Checks contributions for conflict with IPR goals and objectives.

- **Modify**
  If there are patent issues, IPR Dept can often suggest alteration of a proposal to allow for contributions. *It is extremely rare a contribution proposal is fully rejected!*

- **Document**
  Track all submitted contribution proposals and their status.

*Do not Self-Censor!*
Try it, and if (when!) getting an approval from Open Source Board = IPR is cleared to contribute.

Open Source Processes

- **2 Corporate Directives**
  - Intake – Open Source Software Directive
  - Output – Open Source Software Contribution Directive

- **Open Source Board** – Decision making body for contributions of Open Source. Maintains the Open Source Directives. Members:
  - Legal
  - IPR
  - Open Source Operations
  - Technology/CTO
  - Development (GSM)
  - UX (App Software)
OSS Intake

Guiding principal:
In line with intake of 3rd party commercial SW components

Classification of Contributions

**Trivial contribution**
A Trivial contribution is a rather small and simple change to already existing open source software. No new functionality added. No significant improvement in performance.
• E.g., bug fixes

**Medium contribution**
A Medium Contribution is larger than a Trivial Contribution. It can contain both substantially changed functionality and completely new functionality.
• E.g., new Linux device drivers (support for new hardware)

**Major contribution**
A Major Contribution involves substantial amounts of code, with significant Intellectual Property value, and is the result of considerable internal development efforts.
• E.g., DRM Framework
Scheme on OSS Contribution
- Trivial contribution

Note that Trivial contributions only requires Business Manager’s approval!

Scheme on OSS Contribution
- Medium contribution

Representation in the Open Source Board
- Technology
- Software
- GSM
- Legal
- IPR
- Open Source SW Operations

Yes

No

1. Contribution Proposal

Accepted

No

2. Approve / Reject

Sony IPD Parental Approval

If Android

Business Manager / Contribution Officer

Open Source Board

CTO / Executive Management

Open Source Software

Sony IPD Parental Approval

Business Manager / Contribution Officer

Patent Clause?

Yes

No

1. Contribution Proposal

2. Approve / Reject
Scheme on OSS Contribution
- Major contribution

1. Contribution Proposal
   - Sony IPD Parental Approval
     - Patent Clause?
       - Yes
       - No
       - Accepted
       - No
   - Business Manager / Contribution Officer

2. Assess and recommend
3. Approve / Reject
4. Notify

CTO / Executive Management

Representation in the Open Source Board
- Technology
- Software
- GSM
- Legal
- IPR
- Open Source SW Operations

Conclusions

- Engineers empowered, but need to take responsibility
- Legal and Development/Engineers need to work together
- Open Source is simple –understand the consequences and handle it accordingly
- Process/Administration
“SONY” or “make.believe” is a registered trademark and/or trademark of Sony Corporation.
Names of Sony products and services are the registered trademarks and/or trademarks of Sony Corporation or its Group companies.
Other company names and product names are the registered trademarks and/or trademarks of the respective companies.