Taxonomy for
Open Source Software Development

Case Studies on Open Source Software Development Projects at SRA Inc.

Kumiyo Nakakoji
Yoshiyuki Nishinaka
Kouichi Kishida
Yasuhiro Yamamoto

SRA Key Technology Laboratory, Inc.
Nara Institute of Science and Technology
Outline

1. Open Source Software Development
   – the labeling problem
2. Case Studies
   • Wingnut
   • PostgreSQL
   • Linux
   • Jun
3. Comparisons
Open Source Software

- a software program that has its source code open
  - usually under some licensing
    - GNU
    - OSI
    - BSD
- no rigid definition exists
- overloaded meanings
Is Open-source Really a Good Thing?

• the cathedral and the bazaar
• arguments around …
  – security
  – evolvability
  – robustness
  – quality
  – community development
• no “winners” yet because …
What Are They Talking About?

• “Open source” may mean very different things
  – e.g. GNU software versus Linux driver software
The Labeling Problem

• People refer to very different things by calling a variety of software all “open source.”
• We need a better taxonomy for describing, promoting, and assessing open source software development projects.
A Case Study at SRA, Inc.

- SRA has been a leading company in open source community in Japan.
- A case study comparing four projects: (funded by JST, METI, Spring 2001)
  - GNUWingnut
  - PostgreSQL
  - Linux support
  - Jun
- Comparisons, Models, Suggestions
1. The GNU Wingnut Project

- GNU (GNU is Not Unix)
- Richard Stallman at FSF (Free Software Foundation)
- programs as “scientific knowledge to be shared among mankind”
- the “correct” program for a piece of functionality
- centralized control
- strict coding rules, format guidelines, etc.
- business opportunities:
  - porting
  - revision and refinement of feedback to the central authority
2. The PostgreSQL

• originally a research prototype
• database
• strict centralized control
• discussions over a specification (not necessarily on code)
• source is kept open so that people can more quickly find what is wrong in the source.

• business opportunity:
  – customization and adaptation
3. The Linux Support Project

- Linux excluding the kernel development
- “Bazaar” --- distributed control
- multiple versions for a single functionality
- no centralized repository
  - put on the Web and Web search engines may find it
- directory services are necessary

- business opportunities
  - distribution-packaging
  - find up-to-date information over information seeds scattered all over the world.
4. The Jun Project

- VisualWorks Smalltalk application library component
  centralized decision making
- continuous evolution
- reference model

- business opportunities:
  - application development
  - consultation
Evolution of Jun (1)
Evolution of Jun (2)

(1) incremental growth

Smalltalk ▲ ▲ ▲ Smalltalk

(2) reference model development

Smalltalk → Java → C++

time
Jun Evolution (3)

phylogenetic evolution

ontogenetic evolution

3D-Geometry modeler

Version004

Version016

Version340

+ 3D-Topology modeler

8/96: Project HQL started

9/95: Jun was released

7/99: Project NSN started

+ Multimedia handler

Legend:

Community contribution (requests and bug reports)

Refactoring by the Jun developers

Accommodating to needs caused by large projects
## Summary of the Comparisons

<table>
<thead>
<tr>
<th>domain</th>
<th>control</th>
<th>feedback</th>
<th>versions</th>
<th>decision making</th>
<th>motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNU Wingnut</td>
<td>central</td>
<td>MUST</td>
<td>single</td>
<td>vote</td>
<td>reference model</td>
</tr>
<tr>
<td>PostgreSQL SQL</td>
<td>central</td>
<td>bug report</td>
<td>single</td>
<td>vote for spec</td>
<td>fast fixes by “many eyes”</td>
</tr>
<tr>
<td>Linux</td>
<td>distributed</td>
<td>not necessary</td>
<td>many</td>
<td>arbitrary</td>
<td>timely development</td>
</tr>
<tr>
<td>Jun</td>
<td>central</td>
<td>desirable</td>
<td>some</td>
<td>controlled</td>
<td>reference model</td>
</tr>
</tbody>
</table>
Community

GNU Wingnut
- Expert developers
- Core members
- Regular users

Linux
- Core members
- Regular users

PostgreSQL
- Major developers
- Core members
- Regular users

Jun
- Project leader
- Team members
- Regular users
Evolution

GNU Wingnut

feedback

patch

feedback

incorporate

PostgreSQL

Linux

patch

patch

patch

incorporate

incorporate

Jun

released public versions

test versions

9, May, 2001
Three Types Identified

- **Archetype**: as a reference model
  - GNU software
  - Jun
  - for mankind
- **Security-focused**: for fast fixes
  - PostgreSQL
  - by many eyeballs
- **Rapidity-focused**: for timely development
  - Linux (excl. kernel)
  - by many hands
Necessary Issues to Address

• support for documentation
• support for investigating software patent
  – licensing requires to make sure that algorithms used are not patented.
• code assessment by a third party
• SE technology on code manipulation
  – code metrics, filtering, …
• learning program reading versus programming
• incentives for open source
  – governmental support
  – pride, feeling of participation, community, …