

Free Software Communities Management

Jesus M. Gonzalez-Barahona

jgb@gsync.es

<http://identi.ca/jgbarah> <http://twitter.com/jgbarah>

GSyC/LibreSoft, Universidad Rey Juan Carlos

Seminar con Community Management
Universidad Rey Juan Carlos, November 16th 2010

GSyC



Universidad
Rey Juan Carlos

GSyC

(cc) 2010 Jesus M. Gonzalez-Barahona. Some rights reserved.
This work licensed under
Creative Commons Attribution-ShareAlike 3.0 Unported License.
To view a copy of full license, see
<http://creativecommons.org/licenses/by-sa/3.0>
or write to
Creative Commons,
559 Nathan Abbott Way,
Stanford, California 94305, USA.

- Four freedoms:
 - To run the program
 - To study and change the program
 - To redistribute copies
 - To distribute copies of modified versions
- Formulated in 1983 (Richard Stallman)
- Tens of thousands of projects, worldwide
- Term “open source software” coined in 1998
- Different motivations: ethical, practical, etc.

..and free software communities

- Mature free software projects usually have surrounding communities
- Some of them dated in 1980s: first Internet-mediated communities
- Some of them longing for more than 20 years
- Self-conscious communities, developing own infrastructure
- Set of tools evolving over time
- Different actors: volunteers, hired professionals, etc.
- Different roles: developers, bug reporters, support providers, documenters, lurkers
- Different sizes: from less than 10 to 100,000s
- Usually worldwide, with English as common language

Tools for coordinating communities

- First tools (pre-web, 1980s):
 - Mailing lists and News (discussion groups)
 - Release repositories (FTP)
- More specific tools (1990s):
 - Bug tracking systems (GNATS)
 - Source code management systems (CVS)
 - Chat: IRC channels
 - Web site as entry point
- Forges (late 1990s, 2000s):
 - SourceForge launched in 1990s
 - Others: Google Code, GitHub, Launchpad, GNU Savannah, etc.
 - Centralized services for free software communities
 - Source code management, issue tracking system, releases repository, mailing lists, forums, etc.
 - More and more tools available

Organization of communities

- Small:
 - simple, non-written, usually personalist rules
 - mailing list archive records uses, discussions
 - can grow, can disappear, can stay
 - usually in public forges
- Medium-size:
 - strong community culture
 - usually some written-rules
 - may develop and document some procedures (eg: coding standards)
 - may maintain some specific tools
- Large:
 - strong community culture (and sub-cultures)
 - very complex (different motivations, different backgrounds)
 - written-rules, documented processes
 - usually, own forge infrastructure
 - institutional infrastructure (eg: foundation)

Different styles of community organization

- Benevolent dictatorship (Linux)
- Direct democracy (Debian)
- Representative democracy (to some extent, most projects)
- Representative democracy with privileged actors (some projects started by companies)
- Directed community (MySQL)

For each of them, community management is quite a different task

- Small/medium projects: community management is not different from project management
- Free software communities are targeted at producing software: performance metrics, meritocracy
- Long history, quite interesting for study
- Centralization paradox: distributed community, centralized tools (changing?)
- Communities can (and do) build the tools they need
- Interesting models for other communities

To probe further

- Introduction to Libre Software (Spanish, English)
<http://curso-sobre.berlios.de/introsobre/>
- Materials of the URJC master's program on libre software
<http://master.libresoft.es>
- Producing Open Source Software
<http://producingoss.com/>
- Open Sources 2.0: The Continuing Evolution
http://commons.oreilly.com/wiki/index.php/Open_Sources_2.0
- Open Sources: Voices from the Open Source Revolution
<http://oreilly.com/catalog/opensources/book/toc.html>