Open Source Communities

Sebastian Rahtz

OSS Watch

What challenges do open source projects face?

Projects *tend* to start small, and often go in one of seven directions:

- stay small: remains a nerd tool
- gather users but no new developers: frustrated users
- fragment when primary leader loses interest: unattractive for new people
- develop power but with minimal documentation: no way to find the power
- grow within an expert community: high price for admission
- go commercial: stops being ‘free’
- simply die

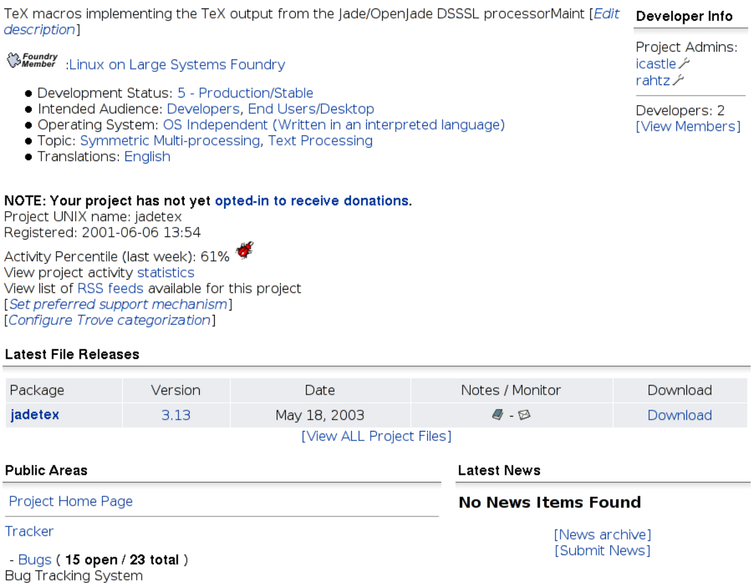
The usual panacea is ‘you need to build a community’

OSS Watch

The reality

Open Source Communities

Sebastian Rahtz



TeX macros implementing the TeX output from the Jade/OpenJade DSSSL processorMaint [*Edit description*]

Foundry Member :Linux on Large Systems Foundry

- Development Status: 5 - Production/Stable
- Intended Audience: Developers, End Users/Desktop
- Operating System: OS Independent (Written in an interpreted language)
- Topic: Symmetric Multi-processing, Text Processing
- Translations: English

NOTE: Your project has not yet opted-in to receive donations.
Project UNIX name: jadetex
Registered: 2001-06-06 13:54
Activity Percentile (last week): 61%
View project activity statistics
View list of RSS feeds available for this project
[*Set preferred support mechanism*]
[*Configure Trove categorization*]

Developer Info

Project Admins:
icastle
rahtz

Developers: 2
[View Members]

Latest File Releases

Package	Version	Date	Notes / Monitor	Download
jadetex	3.13	May 18, 2003	-	Download

[View ALL Project Files]

Public Areas

Project Home Page

Tracker

- Bugs (**15 open / 23 total**)
Bug Tracking System

Latest News

No News Items Found

[News archive]
[Submit News]

Why do people work on open source?

The desire to learn technical skills by joining an open project is strong. Typical reasons for staying in OSS are:

- seeking recognition: 12%
- improving skills: 32%
- improving software: 24%
- ideology 31%

A lot of people work on open source because they are paid to by their company. It isn't all about ideology.

Types of open source communities

- Developers who want to work on a shared project
- Users of software who want to get better results from it
- A community with a problem which is solved by software
- Programmers who find it gets software developed faster
- A procedural infrastructure to provide long-term stability

Communities of practice

from the world of learning theory, COPs are:

- informal networks that emerge from a desire to work more effectively or to understand work more deeply among members of a particular speciality or work group.
- small groups of people who've worked together over a period of time and through extensive communication have developed a common sense of purpose and a desire to share work-related knowledge and experience.
- communities of apprentices where newcomers learn by gradually going from peripheral participation to full participation in the community.

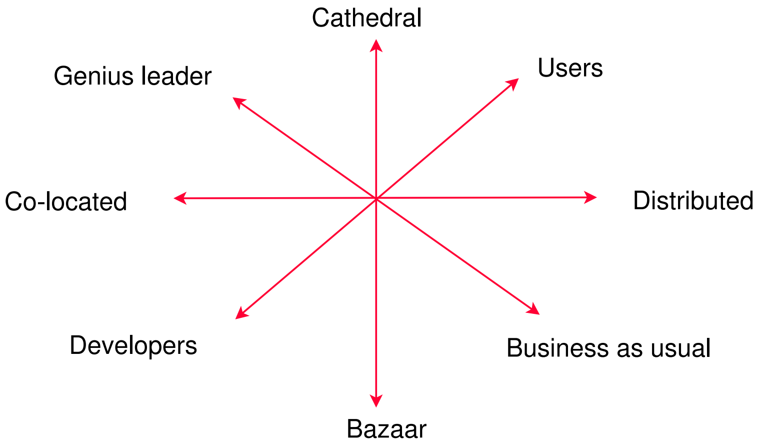
Sound familiar?

- Learning is presented as a process of social

Caveats on the nature of communities

- there are communities of users and communities of developers
- what differentiates open source communities from user groups?
- how are they different from advocacy groups?
- what about open content groups like Wikipedia
- is community the only way of developing good software?
- you can’t control the community because of the OSI licence
- can you have several communities for the same bit of software?

Community tensions



Shining examples of community

Apache web server (etc) a technical strategy; formal democratic management; enviable reputation for quality

Firefox web browser Committed non-technical evangelists and plugin writers

uPortal portal framework shared development between academia and business

Debian Linux Articulated policies and procedures; no prima donna leaders

Moodle VLE a community of teachers which merges seamlessly into developers; charismatic leadership; self-hosting

The Apache Software Foundation

Consists of

1. Projects and sub-projects
2. Project Management Committees
3. The Foundation

The board manages and oversees corporate assets; allocates corporate resources to the various projects

9 members are elected each year but make no technical decisions!

ASF projects

Geronimo Incubator James Logging Maven Perl SpamAssassin Web Services XML Graphics HTTP Server Ant APR Cocoon DB Directory Excalibur TCL XMLBeans Forrest Gump Jakarta Lenya Lucene MyFaces Portals Struts XML

ASF principles

It's a technical meritocracy, with the roles of:

The user
: is either passive, ie anybody who uses the software but does not contribute directly to the project (a 'lurker'); or active, ie anybody who contributes directly to the project

The committer
: An active user elected for merit, who receives this receives an @apache.org account, write access to project repositories, binding vote, the ability to propose others for committership

The member
: a committer elected for merit in the evolution of the foundation, who is a shareholder of the foundation, can propose committers for membership, can elect the board and be a candidate, and can propose new projects for incubation

Firefox

Selling quality: ‘The award-winning Web browser is better than ever. Browse the Web with confidence - Firefox protects you from viruses, spyware and pop-ups. Enjoy improvements to performance, ease of use and privacy. It’s easy to import your favorites and settings and get started. Download Firefox now and get the most out of the Web.’

A community of *users*

Debian characteristics

A free linux distribution:

- multi-architecture (11 architectures)
- very large (1000 developers, 10000 packages)
- volunteers (although some do it as part of their real job)
- distributed (Europe, America, some Australia, not much in Asia or China). European developers seem to tend towards being loner volunteers, in US people are from companies

Debian infrastructure

documents

- Debian Social Contract; Debian wants to move to a social contract which says "free" not "free software"
- policy documents
- developers reference

forums

- IRC for direct access, good for answers
- mailing lists (100+, covering user, developer, announce), good for discussion
- bug tracking system, for archiving
- developer web portal
- package tracking system
- face-to-face meetings, eg at conferences

Debian constitution and officers

- developers
- technical committee (meritocracy, appointed but not removeable; resolves technical disputes)
- project secretary (handles votes)
- project leader (yearly election by developers; talks at meetings, represents Debian, talks to companies, motivates people, coordinates work)
- delegates (appointed by leader to do a job)
- release manager
- teams: release, ftp masters, web, new maintainer, ports, security, publicity
- users reach consensus on decisions, on mail lists
- Software in the Public Interest (SPI). US non-profit which owns trademark, holds money, does legal matters

How does Moodle work?

Moodle is a benevolent dictatorship. Some characteristics:

- Over 4000 registered Moodle sites
- Moodle is entirely self-hosting
- There are support forums in many languages
- `http://moodle.com` offers commerical support through Moodle Partners
- Partners pay royalties to The Moodle Trust to further Moodle development

General community developer rôles

The visionary	has the Big Idea & makes the long-term decisions
The leader	makes the medium-term decisions
The programmer	implements the functionality and makes the short-term decisions
The tester	finds the bugs
The apprentice	programmer fixes the bugs
The documentor	writes the manual
The communicator	tells other people how good it all is
The distributor	packages it up for new users to try

How many of these rôles can safely be filled by one person?

Characterizing development communities

Many open source software development groups have common characteristics:

- small groups of people who have worked together over a period of time and have developed a common sense of purpose
- communities of apprentices where newcomers learn by gradually going from peripheral participation to full participation
- collective understanding of the community by its members and accountability to each other
- shared repertoire of languages, tools, artefacts, etc, produced by the community.

Requirements for a formal community

- named rôles
- written rules (remember the OSI licence)
- communication channels
- discouragement of dissent
- leadership
- legalities, copyright etc

Community guidance

1. Are you clear about the kind of community you want to grow and why? Sort yourself out first.
2. When people find you, do they (nearly) instantly have a good grasp of what your project is about and how your community works together? Sort out your communications.
3. Are there multiple entry points into your community and is there a clear path of progression through it? If your community is more valuable than your code, then value it.

Top tip: start with your web presence

1. Tell people in a simple, summary, way what you are up to:
   - This is what the software is supposed to do
   - This is who it is aimed at
   - There are its main features
   - These are the main software or other dependencies
   - Here are some screen shots
   - This is the developer community
   - Here are licence, download and install instructions

   to have a nice mixed community

Community-building methods

- reward contributors by acknowledgement
- communicate your vision (Gnu ePrints implements OAI)
- make your etiquette plan
- be open and democratic (don't make all decisions at physical meetings at the pub)
- have a good flexible infrastructure
- have an IP policy and a clear licence

OSS Watch

oh, and don't forget the T-shirt