

Project Management in Free Software Projects

Martin Michlmayr
Hewlett-Packard
t**bm**@hp.com

- Some claim that there is no management in FOSS
- There is only self-coordination and self-assignment of tasks
- True to some extent – but not the whole picture
- Decrease of coordination requirements is good
- But coordination is still needed

Seriously, what is “Project Management” anyway?

- Leadership
 - Coordination
 - Organization
 - Planning
 - Motivation
 - Strategy
 - Vision
-
- It's important to know ones limits!
 - Project management also important for developers

What is an “open source” project?

- “Open Source” and “FOSS” really just refer to a license
- The classical model: distributed (global), done by volunteers
- Now it's not so clear anymore

A FOUR-SQUARE DEPICTION OF FLOSS ORGANIZATION

| | Distributed | Co-located |
|----------------|-----------------------------------|--|
| Volunteers | Prototypical FLOSS dev eg Perl | 'sprints' and 'hackathons' eg Zope and Apache |
| Non-volunteers | Virtual work teams eg Ximian | Traditional Workplaces eg MySQL |

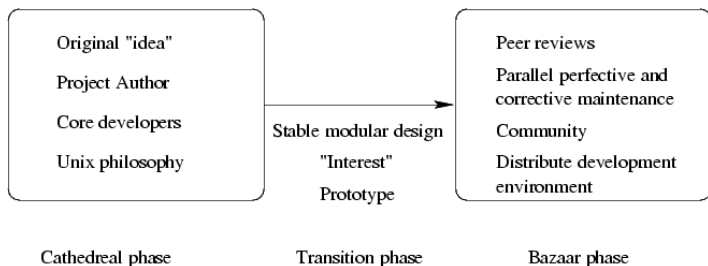
(Figure by James Howison)

Starting a Project

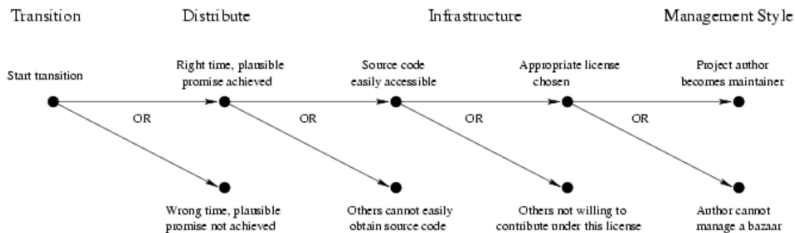
- It's easy to start a project (see GitHub)
- Duplication of effort
- NIH (Not Invented Here)
- Choice of programming language
- Underestimation of effort
- Legal questions (e.g. license)
- Competition might be good

Cathedral vs Bazaar – or Cathedral and Bazaar

- Raymond's Bazaar model
 - Thousands of contributors
 - Peer review
- You cannot start a project in the bazaar
- Logical conclusion: migration from cathedral to bazaar



Transition



You have to:

- attract the interest of others
- give them control
- coordinate
- make it fun!

- Documentation
- Attracting volunteers
- Joining a project, group etiquette
- Infrastructure
- Quality assurance, release management

- Different management styles
- Do coordinate
- Ask people in private, not on mailing lists
- Know your people: what they're good at, what their personal circumstances are
- Getting rid of people (incompetent, trolls, inactive, busy)
- Mentoring people

- Developer/hack sessions: can be effective
- Social/pub: can be motivating
- Downsides: not everyone can participate
- Write minutes
- Do not make major decisions in real life

- Paid people vs volunteers
- How can your project benefit from corporate involvement?
- And how do they benefit?
- Create a win-win situation

Foreseeing Problems

- Growth
- Control
- Dependence on core maintainer: run over a bus, growing up, letting loose
- Inactive and incompetent volunteers

- Exist as a legal entity
- Can receive donations, hold assets, make contracts, etc.
- Transparent
- Governance of organizations independent of one particular individual
- Examples:
 - Free Software Foundation
 - Apache Foundation
 - Eclipse Foundation
 - Linux Foundation
 - Software Public in the Interest
 - Software Freedom Conservancy

- Ask many questions
 - Do you need to start a project?
 - How can you get involved in an existing one?
 - What can you do? What are they lacking?
- Coordination is important
- Self-coordination too
- Look at other projects
- Be a good example!

Culture and community norms

Martin Michlmayr
Hewlett-Packard
t**bm**@hp.com



What is Open Source?

People often speak about THE open source community

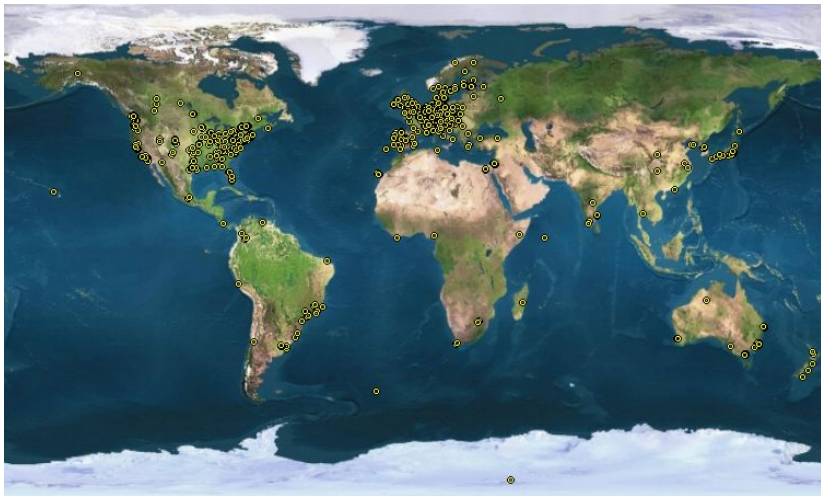
- Technologies
- Infrastructure
- Processes
- Governance
- Philosophy
- Culture



- One of the most popular Linux distributions
- Runs on most architectures
- Has a large number of software applications (packages)
- Is free software
 - Debian Free Software Guidelines (DFSG)
 - Open Source Definition
 - Social contract
- Is completely done by volunteers
 - Open: Debian Maintainer, New Member
 - Some employed by companies
 - Distributed all over the world



Community



Community



Photo created by Alexey Nikonenko

- Community is about being open
- Working with others, including your competitors
- Documentation/NDA
- Communication infrastructure
- Control?



Community has...

- Etiquette
- Norms
- Rituals
- History



Violating community norms is bad



- HTML mail
- Garbled patches
- Long mail footers
- Don't CC people on Debian mailing lists
- Top posting (vs inline replies, but don't try inlining with Outlook)



Date: Mon, 9 Mar 2015 23:52:29 -0400 (EDT)

From: Martin Michlmayr <tbm@cyrius.com>

Subject: Re: my dumb reply

blah bla blah blah blah blah blah. bla blah blah blah blah blah. bla
blah blah blah blah blah blah blah.

> yada yada yada yada yada yada. yada yada yada yada yada yada. yada
> yada yada yada yada yada yada yada yada yada yada

- 1/16: Martin Michlmayr Re: my dumb reply



Date: Mon, 9 Mar 2015 23:52:29 -0400 (EDT)

From: Martin Michlmayr <tbm@cyrius.com>

Subject: Re: my smart reply

yoda wrote:

> yada yada yada yada yada yada. yada yada yada yada yada yada.

blah bla blah blah blah blah blah.

> yada yada yada yada yada yada yada yada yada yada yada

bla blah blah blah blah blah. bla blah blah blah blah blah
blah blah.

- 2/17: Martin Michlmayr Re: my smart reply



Corporate vs Community Culture

- Writing down information / mailing lists
- Distributed development vs co-located development (shouting across aisles, not documenting things)
- Technologies (Lync)
- Use corporate processes internally, and community processes externally



Why Get Involved Upstream?

- It's free software/open source – you can do whatever you want
- Upstream is the authoritative source
- Getting changes in decreases your maintenance burden – in the long run
- Working together, solving problems together
- Getting new ideas



How to Get Started?

- It's hard to give a step-by-step HOWTO
- Use the software, report issues
- Help out, for example by fixing bugs or contributing features
- Not much coordination: you've to work independently
- Basically, find your own niche, do good work and establish trust and a good reputation
- Reputation often stays with the individual



Getting Involved

- Observe mailing lists, study the archives
- Listen and observe for a while before speaking up
- Understand the culture
- Understand how the project works and do not assume it works like a company
- If you make changes, contribute them back (this also reduces maintenance costs)

