Managing A Small Open Source Project

April 9, 2005
WPLUG
Warren Dukes
This presentation is simply one developer's view of how the open source model has worked for one project, Music Player Daemon.

This Presentation includes:
- Opinions
- Observations
- Advice for users and developers
The Example

Music Player Daemon (MPD) allows remote access for playing music (MP3, Ogg Vorbis, FLAC, AAC, Mod, and wave files) and managing playlists. MPD is designed for integrating a computer into a stereo system that provides control for music playback over a local network. It is also makes a great desktop music player, especially if you are a console junkie, like frontend options, or restart X often.
Why Start A Project?

- To satisfy an itch
- To contribute something to the community
- To be famous
- Simply enjoy coding
Why Open Source

- Learn more about coding through peer review
- Can use current open source code
- Learn from open source code
- Can build a community of developers
- The code can potentially survive longer and take on a life of its own
- Looks good on a resume!
  - Potential employers actually have samples of your code
What License To Choose

- **GPL**
  - If someone makes changes to your code and then distributes binaries, they must also distribute the code changes
  - Strict patent limitations

- **BSD**
  - Anyone can change and use the code as they wish so long as credit is given to the original authors
Coding Details

- Choosing which language
  - C, C++, C#, Java
  - Perl, Python, PHP, Ruby
- Choosing dependencies/requirements
- Code Access
  - Tarballs
  - Version Control
    - CVS
    - Subversion
    - Arch
    - Bitkeeper
Documentation

- README
- --help
  - command line parameters and options
- --version
  - name of the program, version number
  - LICENSE, no-guarantee, contact info
- Man page
- FAQ
- HOWTO's
- Wiki
Wiki Example

Main Page

Welcome to the MPD Wiki. Browse this Wiki to learn about the Music Player Daemon and its various clients. Also, you can modify and add to any content. So, please, contribute to this Wiki! :-)

Help Wanted - If you have some extra time, and some knowledge about MPD, you can contribute.

Version 12 - Information about the MPD projects upcoming version 12 release.

For Users

Articles - Articles about MPD
Clients - MPD Clients
Companions - Tools that MPD users may find useful
Configuration - Details of MPD configuration
Dependencies - Information about optional and required dependencies
Publicizing the Project

- Webpage
  - Sourceforge
  - Savannah
  - Your own web site
- Open Source Software Sites
  - Freshmeat
  - IceWalkers
SourceForge
MPD Webpage

About
Music Player Daemon (MPD) allows remote access for playing music (MP3, Ogg Vorbis, FLAC, AAC, Mod, and wave files) and managing playlists. MPD is designed for integrating a computer into a stereo system that provides control for music playback over a local network. It is also makes a great desktop music player, especially if you are a console junkie, like frontend options, or restart X often.

News

mpc-0.11.2 released
March 11, 2005, 2:45 pm
mpc-0.11.2 is out! This includes a few bug fixes and a few new commands: crop, stats, toggle.

ncmpc-0.11.1
February 2, 2005, 3:48 am
Development Cycle

- Development Cycle
  - Specification
  - Design
  - Implementation
  - Testing
  - Fix Bugs
  - Release
  - Feedback
  - Rinse and Repeat

- Open source shortens the feedback loop between Users and Developers
Release

**Major Releases**
- Major design phase
- New Features
- Protocol changes
- Longer Development Cycle

**Minor Releases**
- Bug Fixes
- Minor feature enhancements
- Short Development Cycle
Minor and Major Development Cycles

- Design
- Spec
- Major Release Cycle
- Implementation
- Testing
- Feedback
- Minor Release Cycle
- Release
- Bug Fixing
Packaging

- Source Tarball
  - C* based programs – configure, make, make install

- Packages
  - dpkg
  - rpms

- Distro's and Repositories
  - Gentoo
  - FreeBSD/NetBSD/OpenBSD
  - Debian/Ubuntu
  - Fedora

Let the distribution developers package your code.
Types of Feedback

- **Opinions**
  - Sweet! / Sucks!
  - Ratings

- **Feature Requests**
  - New Features (Users)
  - Design Changes (Developers)
    - “Backend” changes, efficiencies
    - Protocol changes

- **Bugs**
  - Spelling
  - Non-Critical
  - Critical
  - Show Stoppers
Positive Feedback Example

mpd kicks ass

Hi,

I just installed the mpd Debian package, and it looks like it's everything I've wanted in a music server/player. Tomorrow morning mpd is going to wake me up with some sweet tunes. =) [1] So thank you and the other mpd developers for implementing it!

I'll be sure to send you guys patches, if I find something that I need in the daemon.

Cheers,
Juha
Freshmeat Feedback Example

- **Rating:** 8.94/10.00
  (Rank N/A)
- **Vitality:** 0.02%
  (Rank 2523)
- **Popularity:** 3.74%
  (Rank 895)

(click to enlarge graphs)

Record hits: 19,380
URL hits: 12,091
Subscribers: 115

**Comments**

[ »] Wonderful!
by Paiway - Jan 24th 2005 14:11:31

I would just like to say that you people are doing a great job, and that mpd, in conjunction with mpc and gmpc, is by far my favourite music player. Keep it up!
Feedback Mechanisms

- Bug Tracker / Feature Request Tracker
  - Bugzilla
  - Mantis
- Community
  - Mailing Lists
  - Forums
  - IRC Channel
  - Email
- Ratings
  - Freshmeat
  - Polls - Magazines/Forums
Bug Reporting

- Bug Reports from Users are indispensable
  - Leave contact info
  - Give Every Detail that You Can
    - version numbers
    - **config files**
    - explicit error messages / output
    - OS, distribution
- Never simply say X doesn't work!
- Be patient.
Feature Requests

- Understand the purpose for program X
  - Does the feature request match the goals/purpose?
- If only one person requests something very specific, don't expect it to be implemented
  - even if it only takes one line of code
  - even if you supply a patch, the community will still be forced to support this code
- developers time is limited
Coordinating Development

- Documentation! Documentation! Documentation!
  - Keep documentation up-to-date
  - Keeps docs in you version control system
  - Wiki – great for ideas in progress
- Bug Tracker
  - Very useful for delegating responsibility
- Version Control Access
  - Subversion
Community Structure

- Leader
- Board / Committee
- Developers
- Administrators
  - Webmaster
  - Docwriters
- Testers
- Users
  - User and testers who give feedback (such as filing bug reports) are indispensable.
Contributing

- Feedback
  - Bug Reports
  - Feature Requests
  - Opinions
- Testing
- Documentation
- Developers
Coordinating with Other Projects

- Competitors!
  - Share ideas, if possible
  - Share code

- Collaborating
  - Can broaden the user base of your project
  - Can potentially bring in new developers
  - Be open minded!
MPD Collaborations

- Dependency collaborations
  - libao
- Web “Jukebox” collaborations
  - Netjuke
  - Jinzora
- Embedded collaborations
  - IPOD Linux
  - Networked Audio Player
    - Analog Devices Inc. (ADI) demonstrated a networked audio player, running uClinux on the Blackfin STAMP Board at the Embedded Systems Conference in San Francisco.

http://www.linuxdevices.com/articles/AT9272421886.html
Do's and Don'ts

**Users**
- Don't be impatient
  - wait for help
- Do search with google
- Do searches in forums/mailings lists
- Do Give detailed bug reports and feature requests

**Developers**
- Don't be arrogant
  - don't assume you know more about how things 'should' work
- Do propose ideas and promote discussion of these ideas
My Development Philosophy

- I do this for fun
- I have no obligation to implement or anything.
- I care about my users.
- I have a vision of how the things should work and evolve.
- My time is limited, so the things I do spend time on need to be worthwhile.