

# Economics of Open Source Software

Ray Toal

Loyola Marymount University

2006-02-23

Partial support for this work was provided  
by the National Science Foundation's  
Course, Curriculum, and Laboratory  
Improvement Program,  
Award No. 0511732

# Outline

- What is Open Source?
- Why you should know about it (and care)
- Open Source Economy
- Can/Should other fields adopt these principles?
- Future

# You've Heard of Open Source?

- It's been in the news
- Many misconceptions, though
  - not sustainable, pure giveaway, done for fun, detrimental to economy, anti-intellectual property, anti-Microsoft, altruistic
- But big companies are investing in it
- "Much stronger ties to the phenomenon of capitalism than you may have appreciated" -- Bruce Parens

# What's Available as Open Source?

- Linux
- Apache
- Mozilla / Firefox
- OpenOffice
- MySQL
- PHP
- JBoss

# What is it?

- Short, simplistic definition: software for which the source code is, and must always be, even in derived works, available to all
- There are some differences between the *free software movement* (fsf.org) and the *open source movement* (opensource.org)
  - The latter grew out of the former
  - Differences are minor, but basic philosophies the same

# Characteristics

- According to the *Open Source Definition* (v1.9) users of open source software have the right to
  - obtain and utilize the source code
  - freely give away or sell it (bundled or not)
- But they must
  - clearly mark which modifications are their own
  - not place further restrictions on a redistribution
  - not restrict its use

# Other Characteristics

- Open source software is *usually*, but not necessarily
  - worked on by volunteers
  - owned by no one
  - copylefted as opposed to copyrighted
  - free as in speech as opposed to free as in beer (libre not gratis)

# Advantages

- You can always get the source code so you can fix problems yourself
- Someone else can fix problems faster (no waiting months for a next release or patch)
- Benefits of dozens or hundreds of eyes
- Developers are also users
- Security (from massive peer review of the source)

# Disadvantages

# Economics

- Why do people do this?
- Are they sharing? Giving stuff away?
- Can't they get paid for their time and talents?
- Aren't their time and talents scarce resources?

# Basic Idea

- Software is rarely THE product (it usually is for Microsoft, though)
- Less than 30% (5% ??) is sold as product
- Usually it's developed for a customer
- With open source, customers participate in the development - they're not subject to vendors without competition that sell products that aren't exactly what the customers want (and crash too often)

# Giving Software Away?

- Might or might not matter
- Software is really just an enabler (cost-center, not profit center)
- Understand Differentiating vs. Non-Differentiating Technologies
- Don't give away differentiators, but collaborate on non-differentiators (to spread out cost and risk)
- Disincentive to "sit on a patch"

# Non-differentiators

- Available to anyone (all retail software is in this category)
- Works "behind the scenes" as far as the customer is concerned
- Examples: word processors, spreadsheets, browsers, mail clients, web servers, operating systems, databases

# Economic Paradigms

- *Retail* - massive upfront investment (& risk), requires mass market, not customer-directed
- *In house/contract* - customer in control, good for differentiators (unless your contractor keeps rights and resells)
- *Closed Source Consortiums* - history of "titanic failures"
- *Open Source* - started by one group, released early, end-users contribute (no marketing costs, no mass market required, no stock-market investment, distributed cost and risk)

# Contributors (1 of 2)

- Volunteers
  - Want to gain prestige
  - Want to learn about new technologies
  - Want to catch a potential employer's eye
  - Want to gain reciprocity and trust from cooperation

# Contributors (2 of 2)

- Companies
  - May offer two licenses: (1) personal use (unrestricted), (2) commercial (\$\$) to allow inclusion in derived works
  - May sell training and services
  - May sell extra products (IBM sells DB2 on Linux)
  - Hardware companies like open source operating systems and software (hardware hard to duplicate; software is a commodity)

# Commoditization

- When the price of  $X$  goes down, demand for  $Y$  goes up
  - $X$  = flights to L.A.,  $Y$  = price of hotels in L.A.
  - $X$  = cars,  $Y$  = mechanics
  - $X$  = software,  $Y$  = hardware, services, accessories
  - ...

# Open Source is like Capitalism

- No central controlling agency
- Marketplace for software weeds out the useless stuff (many open source products never get used at all)
- Only the best contributions stay in

# Applicability to Other Fields

- Software is soooooo easy to share
- What else is?
  - bandwidth (SETI)
  - music (peer-to-peer)
  - research information
  - biological databases

# Outside of Information Technology?

- Carpools
- What else?

# Future

- Room for both open and closed source? Which will dominate?
- Open source seems to be the only viable economic approach; how long can retailers survive?
- Is Microsoft sending gamers to the Xbox?
- Maybe the idea will catch on elsewhere: social sharing may be a "third mode of economic activity" -- Y. Benkler

# For More Info

- <http://www.opensource.org/>
- [http://www.economist.com/finance/displayStory.cfm?story\\_id=3623762](http://www.economist.com/finance/displayStory.cfm?story_id=3623762)
- <http://perens.com/Articles/Economic.html>
- <http://www.catb.org>
  - Eric Raymond's Essays: The Cathedral and the Bazaar, The Magic Cauldron, ...
- <http://www.fcw.com/article90919-09-26-05-Print>