

What Everyone Should Know About Open Source

Ron Goldman
Sun Labs

Acknowledgments

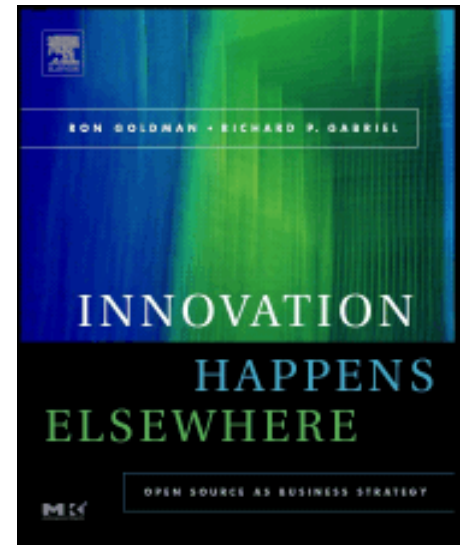
- Thanks to all of the people at Sun and elsewhere who have shared their open source experiences with me.
- This talk is the result of their hard work.

Talk outline

- Introduction to open source
 - What is open source
 - How does it work
 - Building community
- Why does open source matter to
 - Companies
 - Computer professionals
 - Education
 - Regular people

Who am I?

- Advisor to many Sun groups on how to use open source (Jini, JXTA, NetBeans, java.net, OpenOffice, OpenSolaris, etc.)
- Author: (with Richard P. Gabriel) *Innovation Happens Elsewhere: Open Source as Business Strategy* (Morgan Kaufmann, April 2005)
- Researcher at Sun Labs
- Software developer



What is open source?

- Open source is a (new) way for organizations and individuals to cooperate in the development of software
- Much misunderstanding and confusion about open source, especially in the press
 - e.g. open source is much more than just Linux
- Open source has been a part of computing from the beginning
 - SHARE, first computer-users' group formed in 1955 for IBM 701 computer

What is open source?

- Open source = source code + license
- Open source project =
source code + license + community
 - Usually *multiple* communities
- Formal definition by Open Source Initiative:
 - Source code must be available
 - Can modify and create derived works
 - Free redistribution
 - No discrimination

<http://www.opensource.org>

Some well known open source projects

- Apache
- Mozilla & Firefox
- Gnome
- FreeBSD
- Java
- Linux
- OpenSolaris
- OpenOffice
- Perl
- emacs & gcc

Plus hosting sites:

- SourceForge.net 160,346 projects, 1,707,749 members
- java.net 4,448 projects, 388,079 members

Who participates in open source?

- Over 65% are computer professionals
- Over 30% are employees working on open source as part of their job
- Lots of companies participate
 - Sun, IBM, HP, Apple,
 - Oracle, Cisco, GE, Intel,
 - even Microsoft

How does open source work?

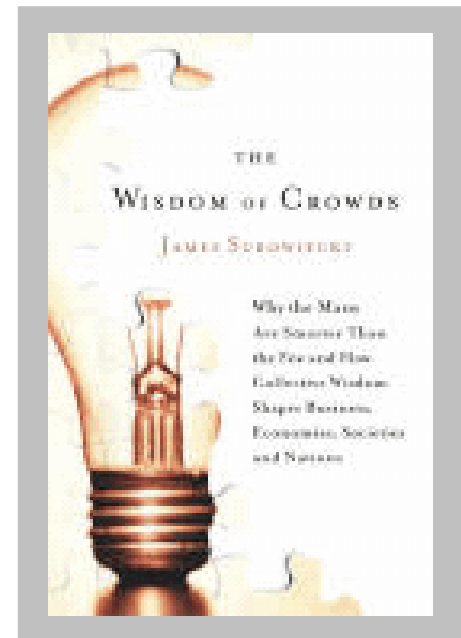
- Copyright owner licenses source code
- Anyone is then free to make any changes they want in *their* copy of the code
- Community decides how changes are made to the *official* codebase
 - Linux: benevolent dictator
 - Module owner as gatekeeper
 - Apache: meritocracy
 - Committers vote on changes

How does open source work?

- Self-organization through shared goals
- Trust enables cooperation
- Achieves critical mass via the Internet

James Surowiecki, *The Wisdom of Crowds*

- Diversity
- Independence
- Way to aggregate results



Open source is an evolutionary process

- Starts with a working piece of code
- Feedback from users
- Developers make (small) modifications
- Community selects changes to incorporate

What the license does

- Grant the right to use and modify the source code
- Grant the right to use IP required by source code
- Sharing requirements for larger works
 - Usually only need to share if redistributing larger work
 - Can it be mixed with proprietary code?
- Other restrictions
 - No warranty, limited liability, termination, ...

Note: License does not describe how developers should work together—it is just a gate one must pass through.

Types of open source licenses

- Can do anything with code—no restrictions
 - BSD, MIT, Apache
- Must only share changes to core code
 - Mozilla Public License (MPL)
 - Common Development and Distribution License (CDDL)
- Must share all changes
 - GNU General Public License (GPL)

Different licenses support different business goals

Contributor Agreements

- Need additional agreement to accept code contributed back into the community
- Contributor agreement states that:
 - Contributor has rights to code contributed
 - Contributor can assign those rights to OS project
 - Copyright assignment (JCA)
 - Patent grants

Open Source vs. Free Software

- Free Software
 - GPL + Free Software Foundation (1984)
 - Free as in freedom, not as in beer
 - Strong philosophy of sharing back with community
- Open Source
 - Term introduced in 1998 to better market the ideas of free software
 - Aimed at being more inclusive of proprietary activities
 - Do not need to share everything
- Both are different from shareware & freeware

Open source is a gift economy

- Commodity economy depends on scarcity
- Gift economy is an economy of abundance
- Gifts form bonds between participants (e.g. a family)
- An individual's status and reputation depends on the quality of the gifts they contribute
- Normal economic concerns motivate using open source, but gift economy dictates how open source community functions

Open source asks “What if what was scarce is now abundant?”

- Don't care if we waste programmer time

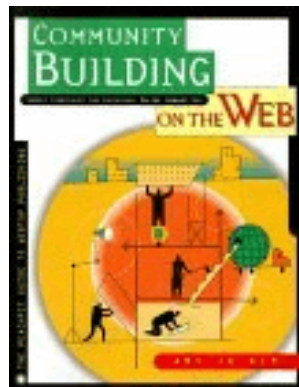
What is a community?

“A group of people who have some shared purpose, interest or goal, who get to know each other better over time.”

Community Building on the Web:

Secret Strategies for Successful Online Communities

—Amy Jo Kim



Note: not the same as a user group

Building a community

- Takes a lot of work
- Need to build up trust
- Need to encourage participation
- Need to create community events & activities
- Project website is natural gathering place
- Trust is built between individuals, not with a company

Encouraging contributions

- Not everyone has write access to the source code archive
- Module owner provides quality control
 - But should not become a bottleneck
- Important to encourage outside contributors
 - Especially at the beginning
- Want to add outside committers & module owners as soon as possible
 - Else risk the community expecting you to do all the work

Who works on what?

- Main development done by core team
- Bug fixes & minor improvements by everyone
- Localization & ports by community
- Planned new features by core team
- Surprise new features by community
- Community development by project leads & community manager / evangelist
- Marketing, documentation, website....

Who works on what?

When the initial core team is all from one company:

- Community generally builds on top of the product
- Major changes to core tend to be left to the core team
- Community acts more like users requesting changes to core

A project needs more than just developers

- Technical writers
- User interface experts
- Graphic artists
- Web site designers
- Technology evangelists
- Marketing people
- Project managers

A company has more resources than a traditional open source project that relies on volunteers – if it takes advantage of that.

Why use open source software?

- It compares favorably with commercial products
 - Quality, cost, features, support, reduced risk
- Can modify the code to suit your needs
- Need to evaluate health of the open source project's community
- Forrester Research survey: about 60% of companies have installed or will install some form of open-source software by the end of 2005

Why open source matters to companies

- Provides new ways to make money
- Lots of business reasons to use open source
- Open source presents organizations with new opportunities to harvest external innovations and to improve relationships with their customers and partners

Open source business models

- Use open source projects as an enabler for other products – OpenSolaris + service agreement
- Require a separate license for commercial use – MySQL or Java ME
- Sell value-adds to the open source version – NetBeans/Eclipse & 3rd party plug-ins
- Integrate and package a solution to sell – RedHat
- Use open source components to lower system cost – GNOME for Solaris
- Disruption of current market – OpenOffice

Business reasons to open source your code

- Development help
 - bug fixes, minor contributions & porting
- Design help
 - early feedback from customers
- Improved quality
 - more & earlier testing
- Support help
 - community members support each other

Business reasons (continued)

- Facilitate technology transfer
 - originators & new developers interact
- Nurture innovation
 - contributions take things in new directions
- Better relations with customers
 - breaks down barriers
- Conversations
 - ideas for new products & product directions

Business reasons (continued)

- Support open standards
 - better interoperability, prevent lock-in of data
- Exploration of a new area
 - get help exploring new area
- Easier to share work with other organizations
 - no need for contracts
- Ubiquity
 - spread the use of your platform everywhere

The costs of open source projects

- Starting a new company sponsored OS project takes additional resources
 - More expensive than proprietary development
 - More success possible / better payoff
- Joining an existing OS project
 - Can be less expensive since massive code reuse
 - Devote company resources to missing/weak parts
 - e.g. GNOME vs. CDE

Some common mistakes companies make

- Not understanding open source (wrong expectations)
- Incomplete or difficult to understand source code
- Users (non-coders) disenfranchised from decision making
- Not adequately discussing plans in advance
- Not marketing project within your company
- Trying to control too much instead of leading
- Not allocating adequate resources
- Not admitting and correcting mistakes quickly

Need adequate resources

An open source project takes more resources than a proprietary one

- Need resources to engage in community conversations
- Need resources to integrate community contributions
- Need resources to follow up on community feedback, e.g. new ideas, new directions

Why contribute back to an open source project?

- So don't pay the “stupidity tax”
- Help shape future development
- To get feedback on ideas
- To establish a good reputation

Why open source matters to computer professionals

- Good way to demonstrate proficiency and establish a reputation
- Companies interested in a OS project will often hire its core team members
- Good way to pick up new skills
- Can continue to work on a project after changing jobs

Why open source matters to education

- Creating a literature of source code
 - Students will have a large body of source code they can study
 - Provides examples of good & bad style
- Easier way to collaborate with other universities and companies
- Courses can directly improve widely used applications
 - Students can work on real-world code

Why open source matters to regular people

- Share tips and practices
- Get help with problems
- Influence future development
- Software freed from commercial pressure now has a chance to become more reliable
 - Windows vs. Linux: which crashes more often?
- Encourages shift to open standards

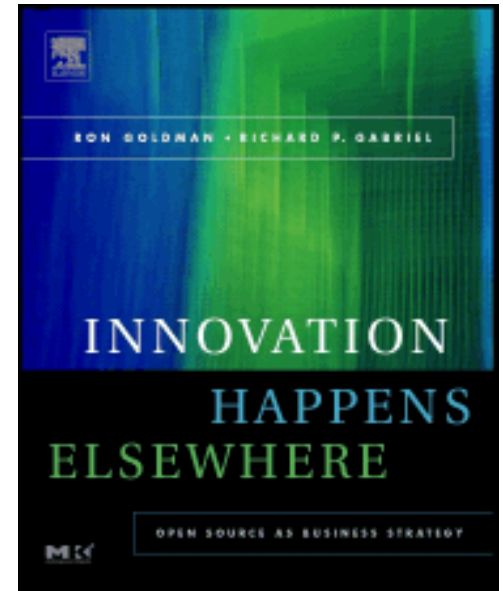
Open source is an evolving methodology

- Current open source projects just the beginning
- Need to explore new ways of doing things
- Users are a neglected resource
- Welcome the unexpected

For further information

Ron Goldman & Richard P. Gabriel,
*Innovation Happens Elsewhere:
Open Source as Business Strategy*,
Morgan Kaufmann, April 2005

Available: <http://dreamsongs.com/IHE/>

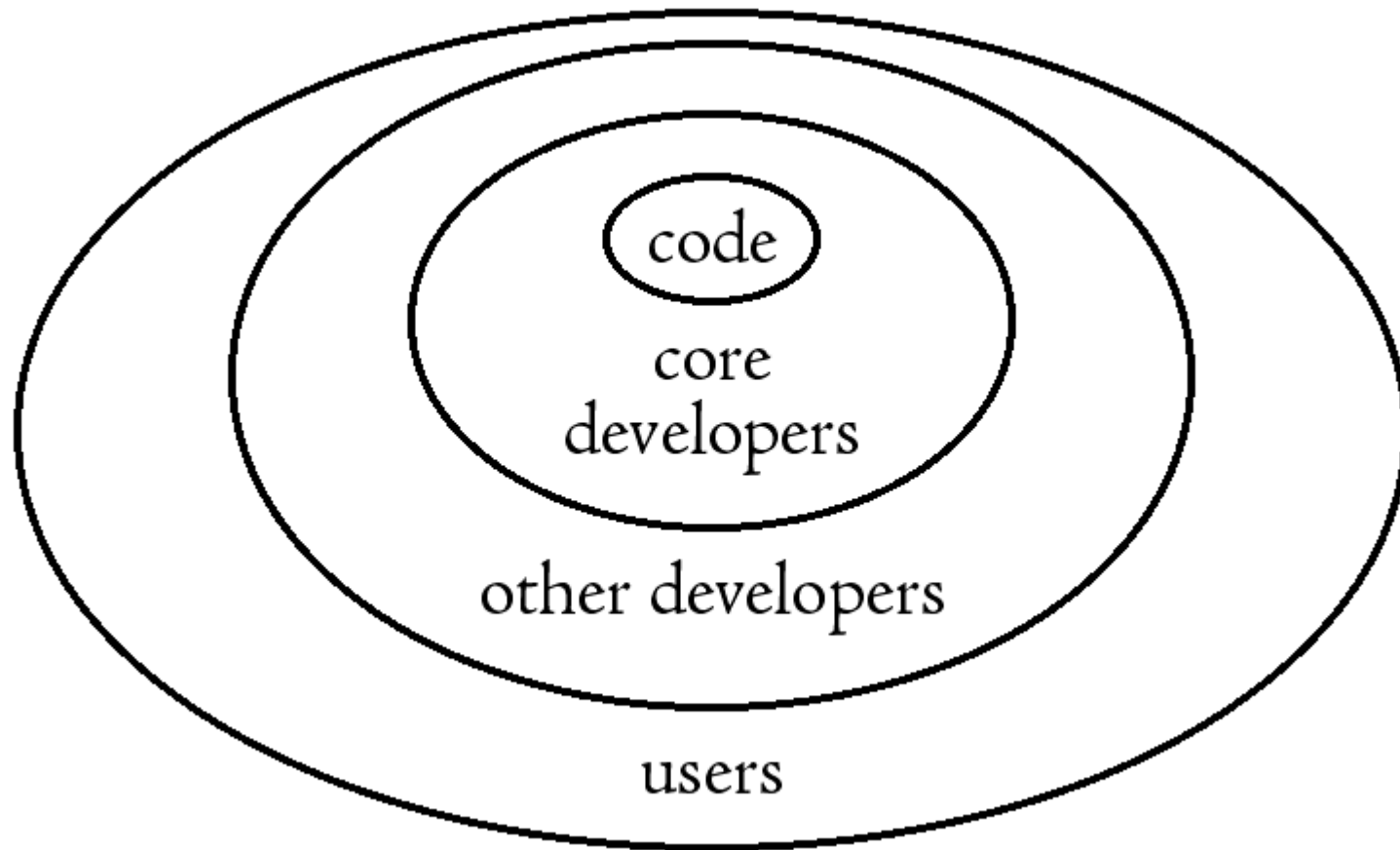


Questions & Discussion

Thank you

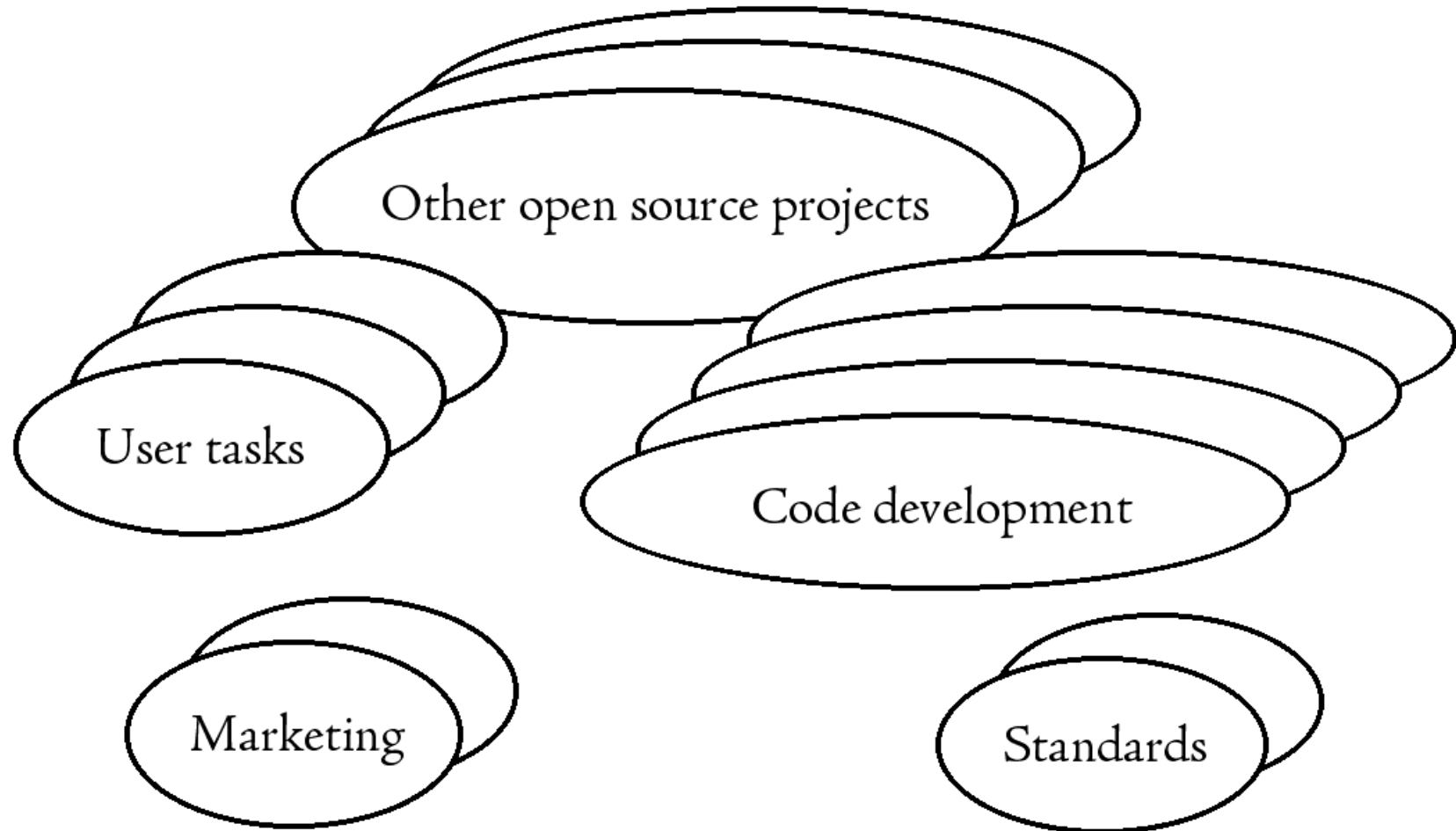
- Extra slides ...

Classic view of an open source project



Single community built around source code

Actually there are multiple communities



Built around common interests

Business reasons (continued)

- Better way to do releases
 - customers can balance new features vs. stability
- Build a market for a proprietary product
 - as open source application spreads so do opportunities
- Commoditize competition
 - free is a hard price to beat
- Statement of values
 - positioning how your company is perceived
- Avoid lock-in
 - open nature prevents lock-in
- Risk reduction
 - reduce dependency on original developer
 - incremental development process more successful

Three principles for community building

1. Design for growth and change.
 - Start small & focused. Grow as needed.
2. Create and maintain feedback loops.
 - Listen to your community
3. Empower your community members over time.
 - Encourage community members to take on new roles.

taken from Amy Jo Kim's *Community Building on the Web*