

OpenSource and Quality

Raju Alluri

<http://raju.spurthi.com/blog/>

OpenSource and Quality: Agenda

- Agenda
 - OpenSource
 - The Cathedral and the Bazaar
 - What to opensource
 - Licensing
 - The Opensource process
 - Engaging the community
 - Establishing the opensource site
 - Quality Involvement
 - Prewrite
 - Adoption
 - Sync up

Opensource

- What is opensource
 - [*Your views here*]

Opensource vs. other paradigms

- Opensource vs. Proprietary (Closed Source)
- Opensource vs. Free Software

The Cathedral and the Bazaar

- Eric Raymond's classic work in mid 90s.
- Some key observations
 - 1. Every good work of software starts by scratching a developer's personal itch.
 - 2. Good programmers know what to write. Great ones know what to rewrite (and reuse).
 - 3. ``Plan to throw one away; you will, anyhow."''

The Cathedral and the Bazaar

- Continued
 - 6. Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging.
 - 7. Release early. Release often. And listen to your customers.
 - 9. Smart data structures and dumb code works a lot better than the other way around.

The Cathedral and the Bazaar

- Continued
 - 10. If you treat your beta-testers as if they're your most valuable resource, they will respond by becoming your most valuable resource.
 - 11. The next best thing to having good ideas is recognizing good ideas from your users. Sometimes the latter is better.
 - 13. ``Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away."

The Cathedral and the Bazaar

- Continued
 - 14. Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.
 - 17. A security system is only as secure as its secret. Beware of pseudo-secrets.
 - 18. To solve an interesting problem, start by finding a problem that is interesting to you.

The Cathedral and the Bazaar

- Key points from the Bazaar paradigm
 - Pick and choose pieces from multiple places
 - Less hierarchy, more helping hands
 - “Compare” the quality of similar products
 - And for you:
 - How do you setup a shop in a Bazaar?
 - How do you attract people in a Bazaar?
 - How different is the delivery process?

What to opensource?

- What are my distinguishing features?
- What is my value add?
 - Common features are “common”
 - It may be worth collaborating with competition
 - For your common base
 - Online retailer examples
 - Invest your resources efficiently on the value adds
 - The notion of secret sauce

What to opensource?

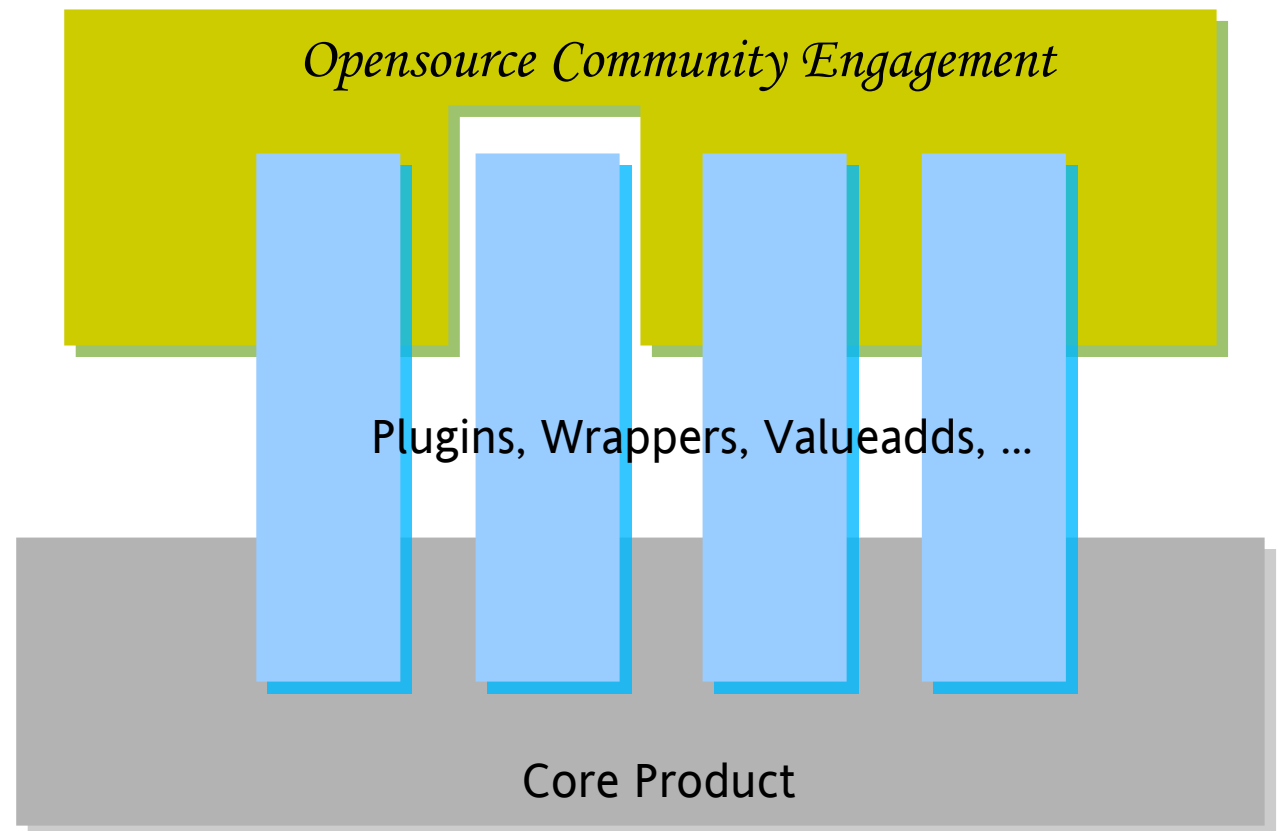
- How about opensourcing entire stack?

Key drivers for opensourcing

- For Corporations
 - Business Model
 - Sales revenues, support revenues and service revenues
 - Core Competency
 - What problem am “I” solving?
 - Resource Investment
 - What to invest in “now”
 - Leveraging Community
 - What can I “get back”

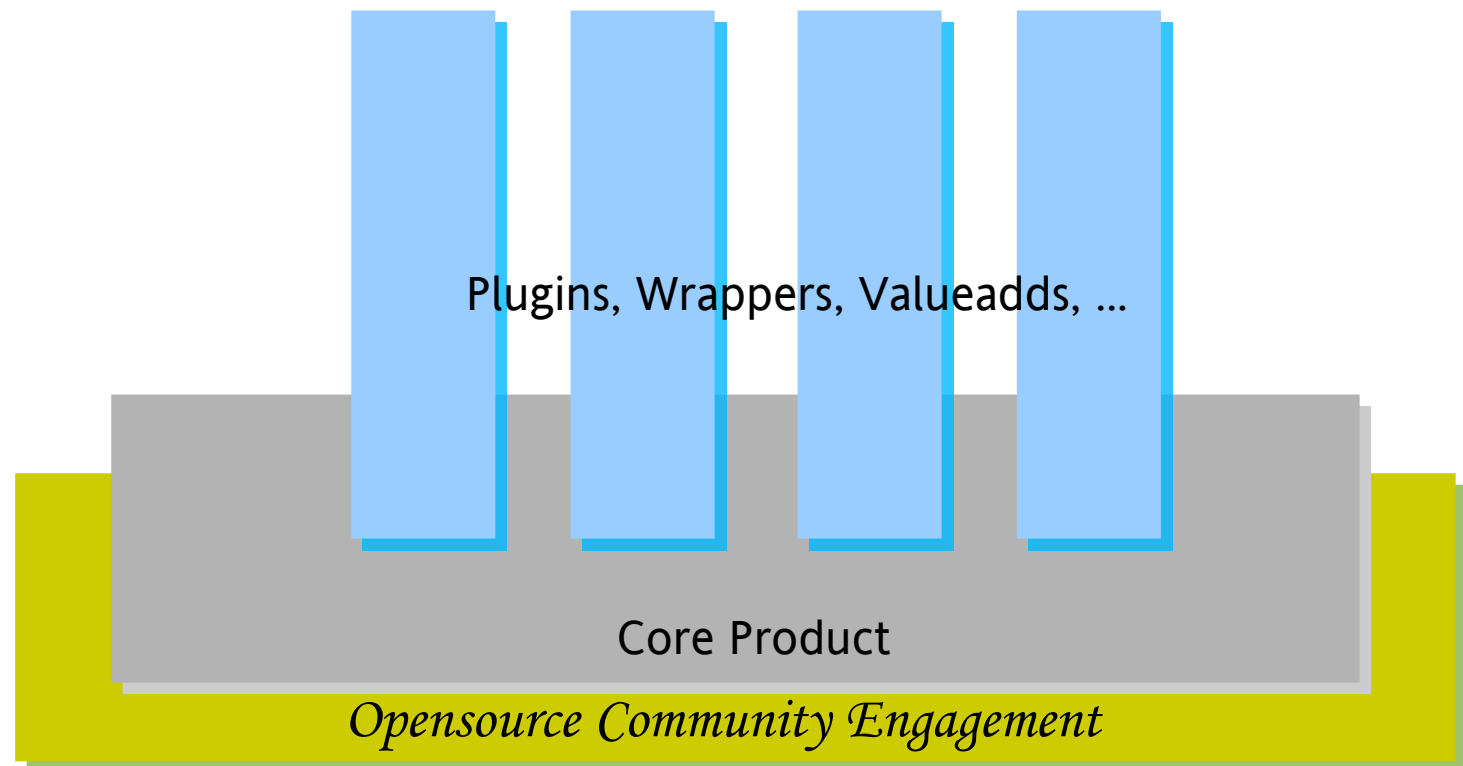
Community Engagement Models

- Scenario 1: Engage community in value adds



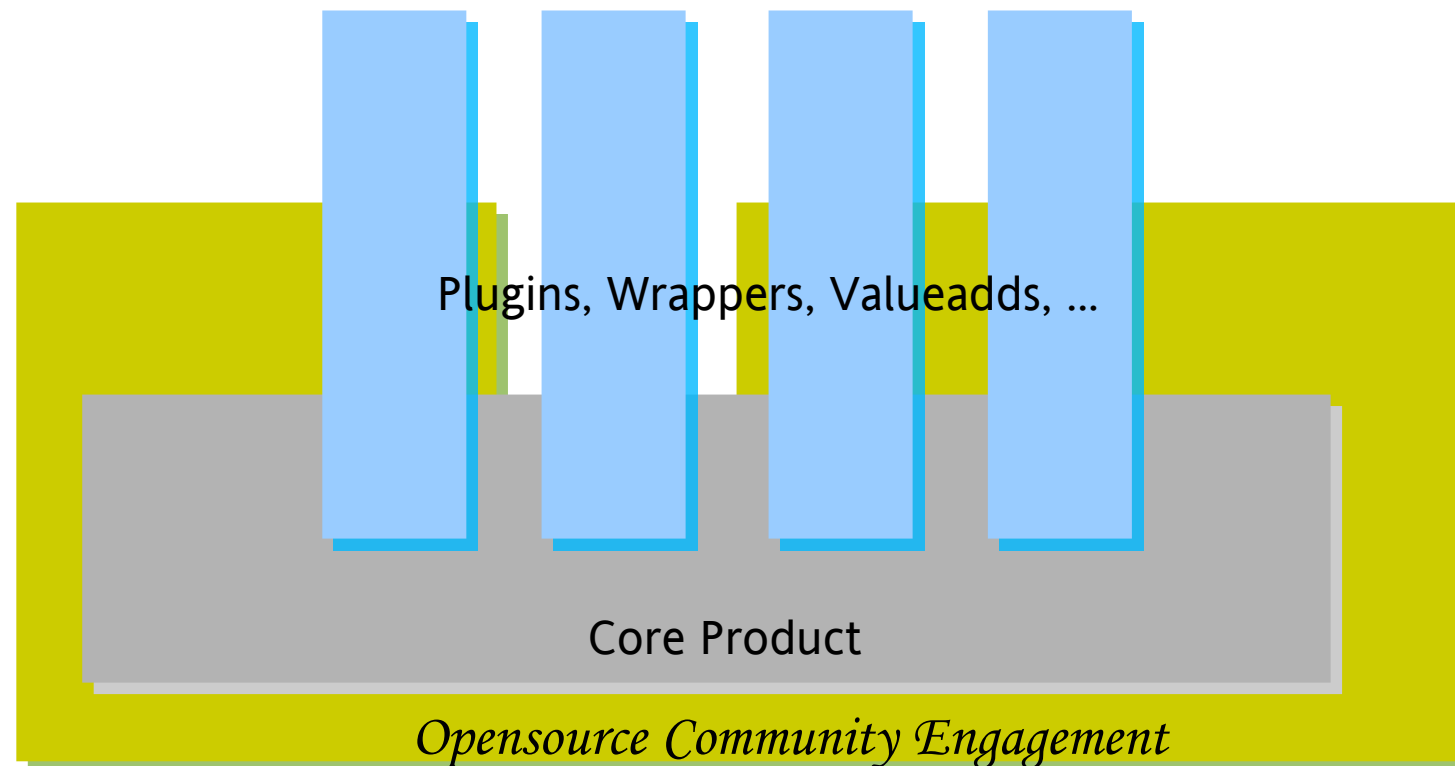
Community Engagement Models

- Scenario 2: Engage community in improving the core



Community Engagement Models

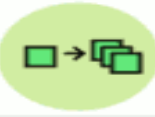







- Scenario 3: Opensource everything, engage community everywhere



Ways to share code

- Licensing
 - Open means many things, when it comes to rights on code
 - Several licensing schemes exist
 - GPL, BSD, CDDL, GPLv3

A quick look at licensing













		EULA	GPL	CDDL	BSD
rights in 'copyright'	 Right to copy	✗	✓	✓	✓
	 Right to modify	✗	✓*	✓*	✓
	 Right to distribute	✗	✓	✓	✓
	 Right to perform	✓	~	✓	~
	 Right to display	✓	~	✓	~
patent rights	 Right to manufacture	✗	✓	✓	~
	 Right to use	✓	✓	✓	✓
	 Right to sell	✗	✓	✓	~

* must use GPL for everything

* use CDDL for things already under CDDL

Courtesy: Chandan B.N.

A quick look at licensing

	GPL	CDDL	BSD
copy	 keep copyright notices		
modify			
	 use GPL for everything	 preserve CDDL	
distribute			
			 can charge money

Courtesy: Chandan B.N.

Typical Opensource Process

- A strategy for the BU or Product
 - Involve Marketing, Services, OpenSource specialists
- An opensource plan
 - Licensing model, website, theme, infrastructure
 - Ways to sync code with community, release models
- Scrub the code
 - Developer code, test cases, documentation, build tools
- Start engaging the community
 - See next slide
- Put a public face
 - Web content, community process, initial members, ...

Engaging the community

- It MUST start even before you opensource
- Several ways to engage community
 - Existing Opensource Initiatives
 - Online forums, faqs and mailing lists
 - Community (even non-opensource) sites
 - Blogs
 - Documentation
 - Customer support channels (internal)

Establishing the Opensource Site

- Decide what is best for your needs
 - java.net, sourceforge.net, custom site, ...
- Decide the content
 - Web pages, mail lists, forums, wikis,
- Decide the code management process
 - Code control (CVS, Subversion, Mercurial, ...)
 - Code change process (projects, developers, submission)
- Related infrastructure
 - User management, downloads, ...

Where does Quality fit

- What are your views?

Where does Quality fit

- A cyclic relationship
 - Good quality helps adoption
 - Good adoption helps quality
- Improved Adoption = Improved Quality
 - Typical Opensource Average
 - 1 error per 2304 lines of code
 - LAMP average
 - 1 error per 3448 lines of code

Adoption: First Quality target

- Improved adoption (of the technology) requires
 - Ease of procuring, downloading, using, ...
 - Ease of building the source
 - Multiple platforms, generic compilers, speed of builds, basic requirements, ease of troubleshooting build issues
 - Ease of installation
 - Predictability, basic requirements tests, informative failures
- Concentrate your test cases on the above

Adoption: First Quality target

- Improved adoption (by the community) requires
 - Ease of using the site (user registration, mail lists, developer processes, bug submission)
 - Simplicity of the site
 - State of documentation (of the product)
 - Ease of executing the test cases
- Evaluating these factors must be part of your opensource plan

Scrub your code

- Are your test cases opensourceable?
- Any licensing issues?
- Any proprietary tools?
-
- Clean them up!

Once the community is engaged...

- What is your quality team's core competency?
 - Core vs. addon, UI vs. backend, Solaris vs. Windows, Large footprint vs. small
- How can you leverage the community for low resource areas
 - “I wrote this nice test case for UI, but if I can get a test case for backend, it will be of great help”

Stay tuned to development

- What does the community develop
 - In next few days? Weeks? Months?
- How can the community (including us) augment to that effort's quality?

Influencing Community

- For fixes you deem important
- For completeness of test cases and suites
- Ongoing process improvement

Ensure Good Sync Up

- With test cases
 - For internal vs. community releases, features, ...
- Internal Processes & Community processes
 - bug processes
 - Filing, tracking, closure, verifying
 - Release criteria
 - Community release criteria vs. Product release criteria
 - Other
 - Code management, Knowledge sharing, ...
- With direction

Few Paradigm Shifts

- Quicker releases to community
- Community release vs. Product release
- What is important to community vs. What is important to Product roadmap
- “Open” ears and eyes

References

- Cathedral and Bazaar
 - <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>
- Chandan's blog entry on licensing
 - http://blogs.sun.com/chandan/entry/copyrights_licenses_and_cddl_illustrated

Q & A