Participating in the Open Source Community

Greg Mancusi Ungaro
Director of Marketing
Novell, Inc.

Peter Bowen
Programmer/Developer
Novell, Inc.

Novell

Agenda

How and why open source communities work

- Myths about open source development
- Benefits to business, customer, developer

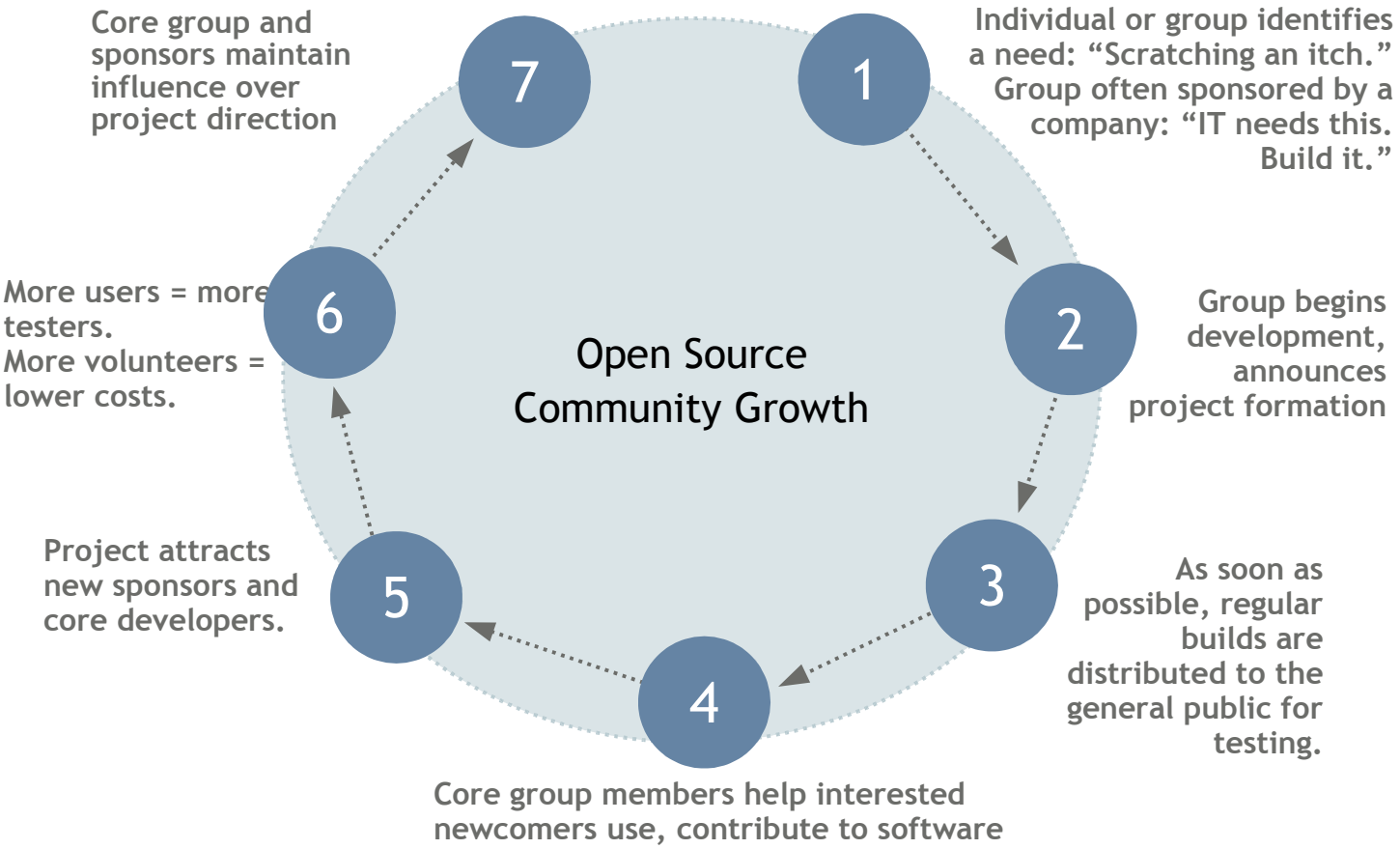
Novell open source experience

- OpenOffice.org
- Evolution
- Linux Kernel and Reiser file system

How you can do it

- When to start, open, or join
- Techniques and pitfalls

Community Growth Process



Novell.

Myths

Myth: Open source development has no costs.

Fact: Open source development can be less expensive than building software alone, but sponsorships, management and participation cost time and money.

Myth: Open source development is totally undirected.

Fact: Some projects are less well-planned than others, but most have long-term plans.

Myth: Open source development is anarchic.

Fact: Development is informal, not ungoverned.

Novell

What's in it for Me?

- Inexpensive informal QA
- Rapid customer feedback
- Inexpensive help from dedicated part-timers
- Reputation as technology leader
- Potential adoption as de-facto standard

Novell.

What's in it for Customers?

Faster iteration and development creates better products, sooner.

Peer-reviewed code puts public reputations on the line, improves software quality.

Open licenses, broader adoption allay concerns about vendor lock-in, “abandonware,” stability of small vendors.

Broader adoption creates horizontal industry platforms, enables user-groups to support each other.

Novell.

What's in it for
My Development Team?

Resource pool now includes global network of experts

Volunteers bring enthusiasm.

Licensing creates enthusiasm: developers won't feel they've 'sold out.'

Recognition and respect from the worldwide network of experts, not just co-workers.

Approaches to Open Source

Starting an independent project:

- More influence, more expense. E.g.: Mono, Evolution

Open existing closed-source software:

- More influence, more expense. E.g. OpenOffice.org, YaST, Mozilla.
- Posting source code doesn't create a community.

Join an existing project:

- Less power, less expense. Larger projects often more successful.

Maintainership

The project maintainer is the acknowledged leader or group of leaders: Linus Torvalds is the Linux kernel maintainer.

Role varies by project, individual style.

Tasks require management skills, but are often performed by developers with no management experience.

Most developers are glad to have someone else do the management part. Companies are good at that.

A company that maintains a project must handle the open source community, business customers, and traditional project management. This is not easy.

Corporate Leadership Requires:

Managing volunteer and community developers is

- Training, supporting, code-auditing
- Volunteers must be persuaded, not directed

Managing interaction with community

- Sort, respond to, feedback and bug reports
- Moderate and support mailing lists
- Host & develop web must supplement product web pages

Traditional project efforts:

- Supplement user testing with formal QA
- Long-range planning
- Note: Final bug-fixing, documentation, translation attract fewer volunteers.

Novell

The Risk of Forking

A fork is a project split, not just a varying distribution

Often caused by political falling-out, developer egos:

- XFree and X.org over development style and licensing

Sometimes caused by differing project goals:

- Galleon and Epiphany browsers over simplicity vs. features

Usually results in the death of one or both branches

- Most distributions now prefer Firefox over Galleon *or* Epiphany

Balancing Community and Cash

Striking balance between stability and development can be difficult: too slow or too fast.

Maintaining “stable” and “unstable” branches can be expensive.

Users of unstable branch may not be target audience, creating misleading market data

Not all users are customers. Demanding users can drain resources without any return.

Novell Open Source Work

Novell Open Source Work

Projects begun by Novell as open projects: Evolution, Mono

Projects begun by Novell as closed projects, later opened: YaST, iFolder

Projects begun by others as open projects, joined by Novell: Kernel, ReiserFS, GNOME, KDE, etc.

Projects begun by others as closed projects, then opened and joined by Novell: OpenOffice.org, Mozilla

Examples: Kernel, ReiserFS

Novell contributions to the kernel:

- SUSE® developers tune, debug, advance kernel alongside others in industry
- Patches submitted “upstream” to primary maintaners (Linus, Alan, etc.)
- Most distributors ship a tuned or edited variant, with their preferred patches included

Novell contributions to ReiserFS

- First Linux journaling file system
- Developed by NameSys and Hans Reiser
- Contributed money and developer time
- Other sponsors include Linspire, DARPA

Novell and OpenOffice.org

Project originally based on Sun StarOffice.

Numerous Novell developers participate, some in key positions.

Community interactions:

- Bugzilla, mailing lists, weblogs, newsletters
- Web page invites contributions, most development broken into individual-sized tasks.
- Appeal to non-developers: art, docs, translations, etc.

How to Work with Communities

When to Open an Existing App

Works best when:

- Highly technical, dedicated user base
- Strong existing need for product in open source community
- Application is not core revenue source
- Application needs new features, more attention from developers and QA

Pitfalls:

- Need to ensure you own all IP in application
- Clean embarrassing comments, bugs before opening.

Novell.

When to Start an Entirely New Project

Works best:

- Strong need for product in the world at large
- No other projects produce similar software
- Service centered on application will be core revenue stream

Pitfalls:

- Need to have strong core and some working code before you can attract volunteer community.
- Similar projects may compete, attract negative attention

When to Join an Existing Project

Works best when:

- Existing project solves some or most of your problems
- Internal developer skill set matches tools, community style of existing project
- Project is large enough to handle new members, small enough to benefit from one new sponsor

Pitfalls:

- Working from outside provides limited power
- Core team may resent newcomers

Community Development Tools

Public and private mailing lists, weblogs, chat, message boards.

Developer weblogs or weblog aggregators

Bug tracking system: usually Bugzilla

Novell.

Community Development Process

Tools must be used

- Keep track of as many bugs as possible in Bugzilla.
- Keep as many bugs as possible public.
- Try conducting meetings on a public chat server.
- Have development discussions on public mailing lists.
- Post meeting minutes to public mailing lists.

Privacy:

- Obviously, don't put customer names, etc. on mailing lists.
- Try not to censor.

Techniques and Rules of Thumb

Start small: assigning ten new people to a small project can overwhelm it.

Expect your developers to spend time on community interaction, building trust and relationships.

Require your team to be polite, even when others aren't.

Be clear about where your interests and the community interests match, and where they do not.

Compromise and humility prevent forking.

Novell®

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

Novell