Abstract: As community manager for a large-scale collaborative project involving over 20 organizations, many of whom compete directly with each other, I have learned a lot of lessons and made a huge number of mistakes. In open source, every person you meet is a resource for wisdom and every experience is an example you can learn from. In this talk, I share simple and not-simple truths about working directly with communities of developers and business leaders in the open source world so that you can learn which examples not to follow.

My handle on IRC is Jefro. I have been embedded in the world of free/libre/open source software since the very early 90s.

Fun fact: my first job was at the first company to offer commercial support for “free software” – the GNU development tools – where I saved the world by answering the phone and delivering mail. I knew how the conference phone worked. The company was called Cygnus Support, and when I joined it there were around 22 employees spread across five or six apartments at a small complex on University Avenue in Palo Alto.
I don’t know how many people in this room are open source experts, or community experts, or open source community experts, or brand new to the whole concept of open source software development and all of the sociological baggage and opportunities it carries. So I prepared a few slides on open source basic principles just in case. By “prepared” I mean I stole them from a talk I gave earlier this year.
Open Source Principles

- Openness & inclusion – open licenses, welcome participation, lower barriers to entry
- Transparency – working without artificial barriers
- Meritocracy – proven worth to the project brings rewards
- Freedom/liberty – no artificial limits on sharing
- Neutrality – all participants are treated equally
- Community – share everything, encourage others to share
- Upstream first – share as first priority, release early & often

Every project has its own methods, processes, and relationships. This talk is specifically about open source projects here, generally software, sometimes hardware or specs or processes. These are projects that offer a license that is open, meaning it is easily obtained, encourages participation, and follows the basic principles of open source outlined here - openness & inclusion, transparency, meritocracy, freedom/liberty, neutrality, community & communication, upstream first, release early & often - to varying degrees. A project’s adherence to these principles is driven by how much the leaders of the project internalize them.

http://www.intelligenceinsoftware.com/feature/it_software_strategy/open_source/#.WKDPyPkrLmE
https://opensource.com/open-source-way
https://opensource.com/node/15018
The 1% rule came from a statistical analysis of wikis and collaborative websites, and in my experience it holds true for mailing lists, bug triage, and pretty much any other collaborative activity - we can just track it very well in technical projects. The numbers are not exact - some people call it the 70-20-10 rule for example - and it is closely related to the Pareto principle, or the 80/20 rule. The concept is the same.

The 100% encompassed here represents the ecosystem that exists related to the project.

In any case, this principle suggests that among participants in a project:

90% are along for the ride and participate passively - this is the user community
9% participate actively, submitting bugs, answering questions publicly - we call these "contributors"
1% help to guide or control the project, assign bugs, determine direction - these are "maintainers", or more simply, leaders

Thus the people who emerge as leaders, or who will eventually emerge as leaders, unless they are founders of the project, follow a logical progression from knowing nothing to being users in the 90%, to being experts and contributors in the 9%, and then hopefully leadership, belovedness, and world domination.

https://en.wikipedia.org/wiki/1%25_rule_(Internet_culture)
Every talk needs a virtuous cycle diagram. This one shows how leadership emerges in a healthy community (open source or otherwise).

One way to read it is that when participation is recognized and encouraged, the person participating cares more about the project, and participates more, because humans crave recognition. But the sentence can start at any point. If I encourage someone to care about a project, chances are higher than normal that they will care and thus participate, earning recognition and therefore encouragement.

I love virtuous cycle diagrams.
A Few Leadership Roles in Open Source Projects

• Technical Leadership “implementers”
  • Technical Director, Chief Architect, BDFL
  • Maintainers
  • Ambassadors

• Governance Leadership “deciders”
  • Executive Director
  • Administrative managers
  • Project member representatives

• Operational Leadership “enablers”
  • Functional systems managers
  • Advocacy & Community

Now let’s talk about these roles. Let’s be clear that roles and people are not the same thing. In some projects every role is held by multiple people, while in others only a few people cover everything. This is a to-do list, not a who’s who list.
And this diagram somewhat awkwardly shows the relationships among all these roles.

If I were Jono I would have some pithy, authentic, wise thing to say about the truths I have learned over 25 years working directly with open source communities, 9 as a community manager. But I don’t. I just hope this is all useful to someone.
Truth

• A belief or idea that is verifiable - verity.
• Another synonym is honesty, or candor.
• My favorite synonyms are authenticity and integrity.

This above all: to thine own self be true. And it must follow as the night the day, thou canst not then be false to any one. - Bill

What do we mean by truth?
Though I am not naturally honest, I am sometimes so by chance.

~ William Shakespeare

Aint that the truth
These are truths and lessons learned around the concepts.
Community is a myth
– or more accurately, a shared illusion

there is no single community - there is a venn diagram of humans
the lesson here is to focus on one group without losing sight of the others
what is community
the relationships between a large number of people
not the people themselves
Competition is in the eye of each competitor

when we first started the yocto project I was very nervous
my open source experience had mostly been in support organizations where the business itself provided upstream support
or in product companies that used and supported open source in a competitive environment
I had been involved with starting an embedded Linux community that was chartered to be available and relevant to everyone
but sadly it failed, for reasons we can talk about when we get to messaging
but Yocto was different. it was real open source, neutrally managed through LF. I joined the mailing list as it was gathering steam in fall 2009, and it was announced by Intel and Wind River at ELCE 2009
and when I started actually working as the community manager the following february we were gathering quotes from 22 organizations, many of whom competed directly with each other
and openly announced cooperation on tooling while they competed at other levels, in support or hardware offerings or professional services
silicon providers, operating system providers, consultancies, product companies
and I thought whoa, this is serious open source
and over 6 1/2 yeas since that sense of wonder has not changed. I have watched companies come and go but many have stayed the entire time
treating the project not like a poker game or a battlefield, but as a commons to benefit everyone
raise the tide
I see companies hang on the fringes contributing little, or even not at all, and get a ton of value out of it
I see some companies go all in to provide people and resources, and still extract more value than they put in - measurable value
in that replacing the project with internal-only resources would result in a lesser solution that cost more
I guess the unvarnished lesson here is that the talk is real, we walk it every day

that doesn't stop the internal politics in many of the member companies whose instinct says this is a cost center
The primary difference between community management and marketing is which way the wind is blowing

where the commerce of a company is made up of countless numbers of discrete soulless transactions, a community is an ongoing conversation between people
talking and listening are equally important

Tracey Erway has run the Yocto Project’s Advocacy team - a good open source term for marketing activities - since before the projet launched
for about the first year we would get on each other's nerves either doing the same things or both of us dropping what we thought the other was taking care of
trying to hammer out the boundaries between my role as community manager and hers as advocacy goddess
what we finally nailed down as the boundary is that the primary role of marketing/advocay is talking - sending messages to the community
while the primary role of community management is listening to the community and responding appropriately, facilitating conversation
this is not a perfect boundary and we still walk on each others' toes from time to time, but I think remembering that helps us remember the great respect we have for each other and it builds compassion into the tiny sub-community that we define as project operations
Expense is different concept from cost

this is a conceptual difference I am making up

- cost is what you give away in exchange for a thing, or a service
- expense is an exchange - more like exercise, where you expend energy to gain strength and more energy
- the idea is that you get more out of it than you put in
- open source communities are like that
- jwz said open source is free if your time is worth zero
- I would argue that if you factor in what you gain from the expense, open source is worth far more than the cost
  - because so much value is created in the process
  - part of that value is in community - in the friends I have in this room and outside
Practical Truths
Politics is nobody's business but everyone's concern

one of the hardest parts of the job is managing people's expectations
which is a political act
I believe that politics is the negotiation of expectations with reality
sometimes through persuasion or argument, sometimes in changing the direction of a
discussion to find more fertile ground for an idea
This doesn’t mean that it is our responsibility to try to control what people do or how they feel
which is often taken as political discourse today
what I mean is quite the opposite
we help facilitate and enable expectations by sharing facts, not opinions, and model
respect and tolerance for the conclusions of everyone in the conversation
Multitasking isn't a real thing
or
You can spend lots of time on the right things
and still get things wrong

there is a joke that half of this job will take 90% of your time and the other half takes the other 90%
this is where discipline is a valuable skill
cm is a gap filling role
that means those of us with a little bit of knowledge fill in the gaps between the people who have a lot of knowledge
on subjects they don't know about, enabling them to focus on the project's core values
and doing it for the whole community around the project
so on a given day I will work on messaging and writing, steer newbies in the right direction, plan meetings and get togethers, give talks, meet with partners, sometimes even file or fix bugs
I drink and I know things
what I find in practical terms is that I do know a little bit about a lot of things
which is the exact opposite skill from a software developer who often knows quite a lot about just a few things
the skill sets are complimentary, but from my perspective it seems like I'm going in eighteen directions
science says that multitasking in humans is not a real thing - that we context switch rapidly, and each comes at a cost
so it is important for personal health to respect that and minimize context switching
because personal health from community leadership drives the health of the actual community
You can spend lots of time on the wrong things and still get things right

I would estimate that I spend about 200 hours preparing for a major event in my industry. I am not an expert at event management so maybe this could be done more efficiently by someone else.

But there are weeks in which I find myself continually on the phone negotiating room prices and finding out how long it will take to ship something to Europe. And I think to myself what the heck does this have to do with managing a community?

At Intel we file monthly status reports, and I often find that I gauge my progress by how often I change the text on each bullet point. Because I have expectations of the things I work on, and of myself and the other people involved.

But expectations are not reality, and at the end, how often does the event crash and burn because someone delivered the t-shirts to the wrong address or there is a typo on the stickers? So far, never.

Because in the big picture, what people get out of events is not directly related to the energy I and others put into organizing and managing them.

People get time with other people. It sounds very kumbaya but it is really just applied psychology, and I am lucky and grateful I get to do it in an atmosphere of incredibly smart people.
This seems like an obvious reframing of negative to positive that's what it is, because humans tend to react to positivity with openness and curiosity, and to negativity with defensiveness
so reframing is a way to make yourself more receptive
you will encounter negative crap in your community
one of the ways I first established myself as a community manager was at the first function I attended as the Yocto Project's community manager
I had been on the job about two weeks
and at this event I was supposed to take notes for a meeting
well, I was in conversation with my boss at the time and I blew it - missed the meeting
the chief architect of the project was not happy and chewed me out with the grace and specificity that only an Englishman can manage.
I was tired and cranky and sort of pissed off that he was making such a big deal of it - but he was totally in the right
I realized as he was speaking that I had a choice in how to respond. My first reaction was defensiveness. I really wanted to say "come on, man, I'm new on the job and my boss wanted to talk to me"
but I didn't. I looked him in the eye and said "you are right, I screwed up. my boss wanted to talk to me and I picked the wrong priority. I'm sorry, it won't happen again."
and then I made sure it didn't happen again - I managed that meeting for the next four years.
luck is having opportunities
wisdom is taking advantage of the right opportunities
You can only prioritize by understanding relative and absolute value

relative value is how tasks compare to each other
people oriented tasks are always, always more important than tasks oriented to objects or processes
and the people should always take precedence
absolute value is how tasks compare to your own values
not project or company values, although you probably will internalize those into your own matrix
understanding your own values is as important as having tolerance for the values of other people. Both are critical maturity points that you need to own to effectively manage a community.
Empathy requires listening to the community
True leadership requires empathy with the community
Therefore, true community leadership requires listening

open source projects succeed on the basis of the health of the community around them
so you might say that
empathy is why open source succeeds
this is what I wanted to end with and I was trying to find a strong pithy jono-like way to summarize the idea into a statement
but the reality is that it seems like an obvious conclusion to me
Did you know there would be a challenge?
Calls to Action

Spend some time writing down your own unvarnished truths, values, philosophies, and practical tips – see where they lead

Challenge yourself with them & don’t be afraid to change them

Share these truths with the community of community leaders
jefro@jefro.net

Jeffrey "Jefro" Osier-Mixon, Intel Corporation