== S1 ==

TITLE + who

== S1 notes ==

A caveat – this is not going to be a presentation full of charts, graphs, and data demonstrating some scientific proof. I hope I haven't deceived anyone as to the theme. We're largely talking in the realm of the social sciences, so instead we point at a few really thick books and say, "This is what those books say, go read them, too."

== S2 ==

* Karsten 'quaid' Wade
** LA born, NorCal raised
** FLOSS advocate for 10 years
** Red Hat Community Architecture team

=== S2 notes ===

My name is Karsten Wade, online I'm known as quaid. If you throw an @ symbol in front of that, you'll find me on identi.ca and twitter.com.

I've been involved in the Fedora Project since the beginning

I've been a free and open source software, or FOSS, advocate for the last 10 years, as I worked in everything from system administration to technical writing. FEDORA CREDENTIALS. For the last few years I've been part of Red Hat's premier community team, which includes former Fedora Project leaders Greg Dekoenigsberg and Max Spevack. We are peers with Michael Tiemann, who founded Cygnus Support 20 years ago as one of the first truly free and open source software companies.

I mention all of that in terms of giving you my bona fides, or why you should trust me. After all, other than the SCALE folks putting me on stage in front of you, what are reasons to listen to me?

The main reason is that the ideas I present today are standing on the shoulders of giants who have been developing these models, methods, and stories for more than 20 years, more like 50, or 100, or 200, or more.

I'd like to ask some quick questions.

How many people here can explain, in 60 seconds, what open source software is?

OK, I'm not going to give you a pat, set answer today. But I hope to widen the thinking for us so we can see how to tailor an answer to each person or audience so it makes the most sense to them. Part of this is being able to talk about open source beyond technology.

Have any of you ever tried to apply the principles of open source to another venue? General business, design, gardening clubs, etc.

== S3 ==

* Being a catalyst in communities.

  To be the catalyst in communities of customers, contributors, and
  partners creating better technology the open source way.

=== S3 notes ===

This is what I'm mainly here to talk about today, being a catalyst in communities, as well as some of the research we've begun to gather, understand, and talk about. Verification of why our community methodologies work, not from a gut instinct level, but from an understanding of the community model that forms around common practices.

Red Hat has a mission statement that is relevant here:

  To be the catalyst in communities of customers, contributors, and
  partners creating better technology the open source way.

Take the carefully chosen word 'in'. This wee preposition is the key to the whole sentence. Imagine how this would sound with some other prepositions: "on" ... "over" ... "of" ... "for " ...

Imagine the classic story of Tom Sawyer, by Samuel Clemens aka Mark Twain, as ol' Tom tricks his friends in to whitewashing the fence for him exchange for paying Tom with various items. Tom gets to take the afternoon off in the shade while his friends do all the work. (Thanks to Chris Grams and his article on opensource.com for this apt analogy.)

This is the model that many people have equated to businesses around free/open source software. Of course, it's the wrong model. It not only doesn't work that way, but that's a sure way to choke off the oxygen to whatever value you may ever gain.

If you call your neighbors together to raise a barn, there is a lot that goes on before you call everyone together. You need to dig and pour the foundation. You'll construct some of the sub-components, and call in your brothers and sisters to help sort materials, and to help build walls. On barn raising day, everyone comes together, lifts the pre-built walls, and works together to rafter the roof to pin the walls together.

You wouldn't call everyone together to work from scratch with a bare field, nor would you wait until it was all together save for the color of paint.

And you definitely wouldn't get away with being Tom Sawyer, directing from the shade of a tree, munching on free apples.

How does this work in reality? Let's take a few examples, and then an anti-example.

== S4 ==

```
--- BEGIN IRC LOG ---
      <rh_pr> We are announcing Red Hat Project! A community-based
              distribution!
  <oss_crowd> rh_pr: Neat.
     <rh_dev> rh_pr: Uh... I'm not ready.
            * rh_pr is away: promoting rhel
  <oss_crowd> rh_dev: what do we do?
     <rh_dev> oss_crowd: I'm not sure.
   <rh_legal> rh_dev: don't do anything until I say it's ok.
  <oss_crowd> rh_dev: what can we do to help with Red Hat Project?
     <rh_dev> oss_crowd: uh... file bugs and help test things.
  <oss_crowd> rh_dev: didn't we always do that?
   <rh_sales> hey, all, if you really want a stable system, don't use
              fedora project. It will eat your brane. Buy RHEL instead.
     <rh_dev> rh_sales: stfu
           --- rh_pr removes voice from rh_sales
  <fedora_us> hey, all, check out our neat community-driven system for
              red hat development
  <oss_crowd> fedora_us: ooooh!
      <rh_pr> fedora_us: I like your name
          --- fedora_rh joined the channel
   <rh_legal> much better
      <rh_pr> We are announcing Fedora Project! A community-driven
              distribution!
  <oss_crowd> rh_pr: Neat!
            * fedora_rh waves
  <fedora_us> I'm not dead yet.
  <fedora_rh> fedora_us: don't confuse things.
  <fedora_us> fedora_rh: does this mean we're merging?
  <fedora_rh> fedora_us: maybe
   <rh_legal> fedora_rh: don't do anything until I say it's ok.
          --- fedora_us joined #limbo
  <oss_crowd> fedora_rh: so, what can we do to help?
  <fedora_rh> oss_crowd: uh... file bugs and help test things.
  <oss_crowd> sigh... didn't we always do that?
  <fedora_rh> oss_crowd: I know, let's all go in the circle and say our
              names.
            * oss_crowd goes in the circle and says their names. This
              lasts several months.
  <fedora_rh> So, there will be the following features in the next
              release of Fedora Core.
  <oss_crowd> Uh... Hold on. Who gets to decide?
   <rh_sales> We do. That stuff will be neato for RHEL-4.
  <oss_crowd> MMkay, then. When do _we_ get to suggest things?
  <fedora_rh> oss_crowd: feel free to talk among yourselves.
            * oss_crowd talks among themselves about new features.
  <fedora_rh> btw, feature X will be disabled in the release.
            * oss_crowd glares at fedora_rh
  <oss_crowd> fedora_rh: nice of you to tell us while we were sitting
              here talking.
     <rh_dev> oss_crowd: sorry, it's just not happening.
  <oss_crowd> rh_dev: when do we get to decide what's happening?
     <rh_dev> oss_crowd: Dunno, I'll ask rh_legal
   <rh_legal> rh_dev: ugh, /msg me
   <rh_sales> rh_dev: let's not do anything rash here.
            * fedora_us gets tired of sitting in #limbo
```

```
  <oss_crowd> fedora_rh: I want to see more of the "community" part of
              the whole "community-based" thing
  <oss_crowd> rh_dev: how about at least a publicly accessible CVS/SVN
              tree?
     <rh_dev> oss_crowd: Yeah, that would be cool.
  <oss_crowd> rh_dev: finally, some movement. When is that going to be
              up?
            * rh_dev is away: talking to rh_legal
            * oss_crowd tries to occupy themselves and do things like
              fedoranews and fedorapeople.
  <oss_crowd> Uh... ping?
  <fedora_uh> oss_crowd: what's up?
  <oss_crowd> fedora_rh: We're feeling kinda useless. What exactly is our
              role, again?
  <fedora_rh> oss_crowd: well, it would be really helpful if you could
              test some things and file the bugs.
  <oss_crowd> fedora_rh: ugh. We ALWAYS did that.
            * oss_crowd begins to wonder what exactly is the purpose of
              fedora_rh
  <fedora_rh> oss_crowd: it's the open-development, proving-grounds for
              new technology component of Red Hat, as opposed to RHEL.
   <rh_sales> Told ya it'll eat your brane.
          --- rh_pr kicks rh_sales from the channel (you're a dolt)
  <oss_crowd> fedora_rh: so, let me get this straight. Effectively, you
              want us to download the packages you release, test things,
              file bugs, and submit patches.
  <fedora_rh> oss_crowd: Sure, why not?
  <oss_crowd> ...but when it comes to things like features, direction of
              the project, and which software to include in the
              distribution, it's the decision of Red Hat?
            * fedora_rh is away: I AM RH
  <fedora_us> I'm still not dead.
  <oss_crowd> fedora_rh: How is that different from how things were
              before the whole "publicly-supported distribution" thing?
  <oss_crowd> rh_dev: where is that long-promised public CVS/SVN repo?
     <rh_dev> dunno, talk to fedora_rh
  <fedora_rh> oss_crowd: look, such things don't happen in a week, ok?
  <oss_crowd> IT'S BEEN A YEAR!
          --- rh_sales joined the channel
   <rh_sales> EAT YOUR BRAAAAAANE.
  <oss_crowd> /mode +b rh_sales
          --- You're not ops in here.
  <oss_crowd> figures
--- END IRC LOG ---

=== S4 notes ===

1. Fedora to RHEL story; community lessons learned from doing that
right and wrong.

2. So we're working on Fedora 13 right now, but back in the Fedora
Core 2 days, things were a bit different.  The addition of the word
core' is a clue -- back then, you had to work at Red Hat to have the
ability to directly put code or content in to the actual image that
made the CD/DVD sets for installing Fedora.  At that time, the NSA
approached Red Hat with a request for help in getting their
Security-Enhanced Linux, or SELinux, code in to the Linux kernel so it
could be included in commercial off-the-shelf (COTS) products.  That's
```

how the NSA likes to do it; they want to help develop ideas initially and keep innovating on them over time, but they don't want to be in the business of supporting software for the rest of the government.

Red Hat made it clear that we could help with the effort, but we couldn't do the job for someone else. One way of helping was by introducing the new code and configurations directly in to Fedora. This included important wider community interaction with Linux users, something Red Hat has experience with and the NSA doesn't. Over the course of a few Fedora Core releases, the scope of work was adjusted in response to the community experience. For example, the initial security policy we tried out was extremely locked down and arrived virtually untested in real world scenarios. It was a disaster that haunts us to this day. The lessons learned there lead engineers to dream up the limited policy, essentially making the entire machine run as before SELinux, and locking down invididual programs one at a time, over time.

Throughout the process, Red Hat played many roles but never one of a single, controlling entity. For example, several programmers from the existing SELinux community were hired by Red Hat to continue that work. Since an explicit goal of the NSA was COTS, and for Red Hat that means Enterprise Linux, there was no hidden agenda. Get SELinux in to the kernel, solve the usability problems, and have it ready for the next forking of Fedora in to RHEL. At that point, the engineers are authorized to do what they already know how to do. They are empowered to help the community make the best technologial decisions. The work from within the communities involved to get the work done. That's how the limited policy was born and tested.

From a commercial standpoint, the result was successful. The SELinux in RHEL 4 was a version of the limited policy, and the policy in RHEL 5 is extremely inclusive and covers nearly all included services.

From a community standpoint, the success was mixed. Millions more people are running more secure systems, but there continues to be a strong ongoing reaction against SELinux. In fact, current statistics from an opt-in system information gathering program called 'smolt' tells us that about 33% of Fedora users do not have SELinux enabled. Since it's the default, they deliberately disabled it.

Why the backlash and lack of usage? It is partially due to the pure nerd nature of the technology involved -- it is hard to put a nice face on it. Another aspect is the way people feel it is forced upon them. That is a lesson from the first days that informs us in other areas -- take care not to decide in private and then force those decisions on the public. In those early Fedora Core days, it was hard for all the developers involved to be entirely in charge of an open sourcing process. While it is late for SELinux's reputation even though it is totally awesome, other technology decisions have gained from this lesson in Fedora.

3. Part of doing this work is making big and public decisions that you have to build upon as much as you wish there were a way to spin back the clock. In the run up to RHEL 5, it was clear virtualization was going to be important. Virtualization is a way of running more than one server on the same piece of physical hardware, with a single hypervisor managing the interactions between virtual host and the

physical hardware.

To make the timeframe for RHEL 5 meant getting code in to Fedora in time to be vetted by the community. That's one of the ways Red Hat ensures that the code in the commercial product is of sufficient quality -- this is the community quality assurance stage. Also, it's really the only way code gets in to RHEL. At the time, there was only one viable free/open source virtualization choice, Xen. Work was done to integrate Xen in to Fedora, and the RHEL 5 release featured Xen by default, enabled to virtualize RHEL 5 and, later, RHEL 4 hosts.

At the same time, Red Hat's virt team realized there were going to be more than one player for Linux and other virtualizations, and they began to work on tools that extrapolate the virtualization into layers. One tool, libvirt, is a library that applications can write to; libvirt then handles the interaction with the virt hypervisor. The idea is, libvirt can be expanded to cover more than one way of virtualizing, but you only have to write your application against the one library and it can run in many virtual environments.

While this was occurring, a few smart community engineers were working on KVM, or "kernel virtual machine". This is a virtualization choice that runs natively in the Linux kernel, and for various reasons, was immediately a potentially better choice for virtualizing Linux. Because the engineers were working directly in the Linux kernel community, that meant that Fedora gained KVM during the regular process of updating kernel versions during the development cycle.

Similarly, RHEL was able to gain KVM as an additional virtualization solution during a RHEL 5 update, which happened to be after it was in Fedora for a few releases. What is really a major sub-system was added in during a regular release update without disrupting existing running systems. This was in part due to usage of libvirt as well as the quality of work done in making KVM part of the kernel, and the quality control experience through Fedora and other Linux distros.

Eventually, Red Hat acquired the company that wrote and maintained KVM. The reason this is sort-of an anti-example is because in this case Red Hat made a decision to pursue Xen when the rest of the Linux community was not sold on the idea. In a sense, Red Hat's will was put first, but only in a situation related to Red Hat's own product family, and not in a way that distracted from other work. Producing libvirt made it easier to adopt other technologies, and perhaps the knowledge that Red Hat was able to adopt KVM when it was mature enough helped to drive that development process.

In this case, Red Hat was an effective catalyst by remaining within the community, separating out and working on business interests, and spending the extra resources and effort to ensure that regardless of the technology the community chooses, our customers and partners can benefit with minimal hassle.

== S5 ==

* Professor's Open Source Summer Experience (POSSE)
** Being a catalyst in education

[POSSE LOGO]

=== S5 notes ===

Let's head out in a different direction now and talk about other meaningful places. Our team has a limit on what we can focus on, and while we are all passionate about education, we have to make some hard choices of how to spend our time. Right now, we feel the best place we can make a direct difference ... place to be a catalyst ... is in the realm of higher education. For a number of reasons, it is harder to move our agenda of teaching involvement in open source at the primary education level. One of our solutions is to work on this at the college level, and havethat work draw in changes on the pre-college level.

Computer science and engineering students are graduating without ever having worked on a real-sized codebase. Without ever collaborating in a way that mimics what they'll experience in the rest of their lives as programmers. Without using the tools and processes that are at the core of development practices, whether the source is open or closed. This is one of the reasons I've heard from Google about why they started the Summer of Code program. They were hiring people out of college and having to immediately retrain them on stuff they should have known already.

This being in involved in open source is different from what we saw much, much more of, which is learning how to use open source tools and how to develop with open source programming languages. Java v. Java.net.

In the course of discussions with educators on how to help solve this, we learned several things. We took a simple scientific approach. Make observations, combine with previous experience, knowledge, and proof, and let these inform our approach:

* There was no single location for discussing and aggregating practices, texts, and tools for teaching how to be involved in open source. There was no community of practice.

* With the way the academic calendar and culture are run, many educators didn't have time or incentive to make participating in open source part of their curriculum. We call this "the chicken and egg problem" - without one, you cannot have the other, but which comes first?

* There is one formula that had the most success: having the educator directly involved in an open source project as a contributor, and that project is used for teaching students how to participate. Who does this? Dave Humphries, Chris Tyler, and associates at Seneca College in Toronto with their relationship to Mozilla and Fedora; Bart Massey at the University of Oregon and Xorg; Steve Jacobs at Rochester Institute of Technology and games for the Sugar/OLPC XO platform. Recently, Matt Jadud and another colleague did an open marketing section of a class at Allegheny College, with the students

  working with Fedora Marketing on modular deliverables for the Fedora
  13 release.

This gave us two clear courses of action.

1. Coordinate with some of those people who successfully teach open
   source and form TeachingOpenSource.org.

   Successful in e.g. ease of creating track/splash at OSCON this
   year.

2. If we could do something that taught educators how to be involved
   themselves, they could run with that in creating classes. The
   Professors' Open Source Summer Experience, or POSSE, was born. The
   first POSSE was held in the summer of 2009 in Raleigh, North
   Carolina.

== S6 ==

POSSE RDU 2009 [pic]

=== S6 notes ==

In POSSE, experts from the free/open source software community come to
teach educators details about how to be involved in real projects.
This is done through a combination of instruction and hands on
actual-contribution activities. One goal is to teach the teachers how
to be 'productively lost'. In other words, to mimic within one week
the experience of being a contributor, with real work outcome from it.
That first POSSE focused on packaging, and they brought in an expert
who also happened to be a high school student. The irony of Ian
teaching PhDs was not lost on anyone. The professors-as-students did
everything from install Linux, get a contributor account, make edits
to the wiki, and join the packaging system.

A second POSSE occurred in the fall of 2009 in the Asia-Pacific
region, bringing together educators from China, India, and Singapore.
This group initially had a hard time with the less-structured learning
approach that works better for free/open source software. Then they
had an experience that gelled things together for them. They were
working with Fedora designer Mairin Duffy, who was back in Boston and
teaching via IRC. For those who haven't experienced it, IRC is a
text-based chat. It takes some skill to navigate and participate in a
disussion. Over the course of an hour, Mairin took them through the
process of checking out, translating, and uploading localizations, or
translations, of pages for the then-upcoming Fedora 12 release.

When Mairin was done, the class took a break. When they got back
together, the instructors asked, "Now, how many of you understood
everything Mairin just told us to do?" Everyone admitted they did not
understand at all. "Neither do we," the instructors said. This
stunned the educators. How could these people teach a class they
didn't know anything about? This was when Mel, one of the instructors
and my colleague in Community Architecture, she wrote on the board,
"productively lost." The class instructors then took the class
through the IRC log step-by-step. They broke out the information and
tasks, showed everyone how to find the resources referenced, and
generally took the group from completely lost to actively producing

translations.

By the time I came a day later to help teach about doing free content documentation, the class was very ready to engage. I was able to take them through editing their first wiki page, helping them to find an actual reason to contribute something to the wiki, and show them how to show other people the same thing, all through IRC from my home in Santa Cruz.

This July 6 to 10 I'll be one of the instructors, along with Mel Chua and OSI Board member Alolita Sharma, teaching in Mountain View, Ca.

== S7 ==

* From POSSE to Communities of Practice

=== S7 notes ===

An interesting thing happened at that first POSSE. My friend Greg Dekoenigsberg was teaching the session, and this is his story I'm retelling.

So Greg's there at the first POSSE, proving the model we've developed over the months, that there is something valuable and new to teach these professors so they can go forth and do great things.

He is teaching something valuable about how the free/open source software communities work, and a pair of the professors pipe up with a comparison. "That sounds like communities of practice," and Matt Jadud from Allegheny College and Cam Seay from NC State University proceeded to pull forward this entire body of academic research about how communities form and grow sustainably.

Here's Greg, he thinks he's imparting something entirely new, and he discovers it has an academic discipline with real PhDs. Naturally, it has to be that way. We already know we're not doing anything far different from a barnraising, right? It's just funny to realize we're struggling to define something, making up terms and rules, and we discover there is an entire discipline around it that wasn't in our view because we'd been immersed in not-academia for so long.

What's funny is that all the researchers were studying this separately for years, and when they discovered each other, they ended up essentially forming a community of practice around studying communities of practice.

Communities of practice is a scientific discipline because of its approach. It is knowledge built up by researching using the scientific method. The scientific method is a clearly progressing pathway that is brightly lit when you look backward, and helps provide illumination as we look to the future. Given the relationship between the freed software methodologies and the scientific method, it feels especially good to rely upon the communities of practice theories. We are practicing our joint methodology -- building on the good work of others to advance the leading edge. It is great to be able to tell Red Hat, "We do this for more than just feeling good and the ethics of doing the right thing." It's great to be able to come to you, the free/open source software communities, and tell you about the fact

that we are using the best method to achieve our collective and individual goals. The community method works, it's proven.

Etienne Wenger, the leading theorist of communities of practice, defines the term as follows:

> Communities of practice are formed by people who engage in a process of collective learning in a shared domain of human endeavor: a tribe learning to survive, a band of artists seeking new forms of expression, a group of engineers working on similar problems, a clique of pupils defining their identity in the school, a network of surgeons exploring novel techniques, a gathering of first-time managers helping each other cope. In a nutshell: Communities of practice are groups of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly.

[CITATION]

Why the science around CoP so important to us? It gives a different measuring stick - it's not about downloads, or how many people are in the account system - that's not how you measure the health of the community. CoP gives us a structure for how a healthy community should look - it's a list of items that should be happening in there, and if they aren't there or are broken, then now you know where to put your energy.

These folks were influenced by another discipline embodied in the book "A Pattern Language". That language was defined by a group of architects who looked at how humans have built their environments, from homes to cities, and what are the patterns that appear over and over again. For example, you let humans form their own walking pathways across open spaces, and when they've done that, that's where you put the permanent footpaths. They looked at how well the patterns interacted, and the whole pattern language is something you can use when you design your own house - it gives you a pattern about how big make a porch must be to be used

== S8 ==

Elements of a
Community of Practice
--
Domain (what)
Community (who)
Practice (why)

=== S8 notes ===

Every community of practice consists of three structural elements, which need to be strong and present or you don't have a CoP:

* Domain. The domain is the area of knowledge that interests the community. In free software, the domain is usually a particular technical problem that needs to be solved.

  - Skateboarding
  - Hunting

- IP Law

* Community. The community is the set of people who care enough about the domain to give their own time to participate. In free software, even though a domain may be very specific, interested community members can come from anywhere that's connected to the Internet -- which is one of the factors that makes the community software development model so powerful.

  - Skateboarders
  - Hunters
  - Lawyers

  The community are the people who are involved, but remember that participation can take many levels. In the "Cultivating CoP" book they tell the story of a company that had a CoP, and one person was always at the meetings, but sat quietly in the back of the room and for nearly two years never made a verbal contribution. Then this person moved to a new department, and one of the first things he did over they was to form a new CoP identical to the first one, from all he had learned while sitting in the back of the room. It shows that you can never know what is going to happen when you open things widely, but you create possibilities. You will be surprised and delighted.

* Practice. The practice is the way that work is done, by the community, to further their goals in regard to the domain. All frameworks, tools, ideas, stories, documents, legal entities, code, and so forth, are all part of the practice. It's the work, and all the tools used to get the work done.

  - Skateboarding tricks, tips, secret locations
  - Hunting tutorials

All of these have to be in place to be a CoP and to reap the benefits. You can have a community of people from a domain, but they get together and drink beer, and never get down to the business of recording their process. You can have a group that shares how-to information/practices, but never works together as a cohesive community.

An important aspect of these elements is that the community is self-documenting. In the process of practicing in the domain, community members capture content about the practice - its history, processes, and so forth.

== S9 ==

Principles of
Communities of Practice
--
1. Design for evolution
2. Open a dialogue 'tween in/out
3. Invite different levels of participation
4. Develop public/private spaces
5. Focus on value
6. Combine familiarity & excitement
7. Create a rhythm for the community

== S9 notes ==

I'll quickly explain the principles for cultivating a community of practice. You may recognize these principles as having practical equivalents in free/open source software projects.

* Design for evolution

  - Breakthrough of live spins came from non-core engineering groups

* Open a dialogue between inside and outside perspectives

  - Clique is expected, make sure it's transparent how to communicate inward and outward.

* Invite different levels of participation

  - This is how we build experts. Rule of thumb - if it doesn't hurt the community, don't stop, encourage them.

* Develop both public and private community spaces

  - It's what makes the open, transparent work. Not just sensitive issues, but building personal relationships.

    - For example, some people approach a group with different caution, or they keep themselves private while learnig about the community. This might be called shy, but rather than label it, realize that this is an opportunity to sit down and have a private dialogue. Through this, the person learns and is made more comfortable about joining in with other, public community activties.

 * Focus on value

  - People need to see what they are putting in and what they are getting out of the experience. In FOSS projects, regular release cycles give us a chance to see the work sooner. Wiki - instant value.

* Combine familiarity and excitement

  - Ex. from CoP book - Prince Street in Boston, old picaresque community, grew organically over time, people live upstairs, businesses are downstairs. Shop doors are set back, maybe with chairs or goods to sell. That invites people to pause in the doorway, have conversations. As you walk down the street, you see side streets and walkways that curve off gently and interestingly. All of these pattern language elements create a sense of familiarity, comfort. But you also know if you go down to your favorite cafe, there's a good chance you'll have an excellent conversation with someone. There is a sense of excitement amongst all this familiarity.

* Create a rhythm for the community

  - We use this in open source software - meetings, releases, etc.

[30 seconds for each principle.]

All of these, the 3 principles and 7 elements, when you see them working in a community, then you know the community is healthy. If your goal in an experiment is to grow something in a petri dish that phosphoreses, then you know you did it when the dish glows in the dark. If you get someone who comes along and says, "Yeah, but I want to know how many microbes are in there, and which are glowing and which aren't, and can we optimize for the glowing ones ..." You do that, and you no longer have a community - you have a dissected your experiment. You have pulled in metrics that don't have meaning for that scenario.

== S11 ==

* Free <3 Open <3 Free

=== S11 notes ===

To decipher that text for you, that means, "Free loves open loves free."

I want to take a moment and talk about the constructed debate about free and open. It has been a common practice to take a debate about terminology within a community of practice and turning it into an A vs. B situation. Journalists do this from the outside, fans do this from the inside. It appears to people even in the midst of the debate that the points under discussion are extremely defining and important.

I work at Red Hat, arguably the world's most profitable pure open source software company. A few years ago we acquired JBoss, which included personnel, commercial products, and the stewardship of a community that largely were also employees. JBoss's practice was hiring anyone from the open community who became valuable in the development team. The Java developers in the JBoss commercial and community space have a pragmatic view about open source software, and they are as likely as not to be the group of fans making fun of the "free hippies".

You see, we have a microcosm of the "free v. open" debate within Red Hat itself. I came clearly from the Linux side of the house, and while I thought the "Free v. Open" debate there was heated, I didn't have a clue until I began to spend serious time with open source Java folks.

Clearly I'm stereotyping here. There is a full spectrum where "more free" and "more open" are sort-of ends, and members of the extended community of practice around free and open source software are scattered across it. But this is just debate, a healthy part of a community. It is hardly truly divisive.

In fact, we often forget that fundamentally we are all in violent agreement. A very high percentage, say 80% or 90%, of the folks in the wider FOSS communities agree about the practice of FOSS. We agree about how we do it, why we do it, why it works, and how you can do it yourself. So, I might disagree with Bradley Kuhn of the Software Freedom Conservancy about exactly how freed Fedora is because

of the inclusion of distributable but closed source firmware in the Fedora release. But we are clearly in the same camp.

It comes down to the power of branding. Remember, a brand is a sponge, and is as much what the rest of the world says it is as it is what you the brandholder say it is.

In Tom Friedman's book, "The World is Flat," Friedman identifies open source as one of his top-ten flatteners, specifically he calls it "the most disruptive force of all." This has in turn influenced many thousands of CIOs to recognize their need for an open source strategy. Much easier to sell an open source strategy to the Board than a free software strategy.

We've seen the effect of this shift in CIO attention. Red Hat has dramatically expanded our business, from where Wall Street in 2003 was a sizeable share to recently where we've had to reassure financial analysts with reminders that Wall Street now represents less than 10% of our business.

While the free software brand has been working great for hackers, the open source brand has been working great for business.

If we had left the branding as "Free as in freedom" along with the stance and explanation that gave us, we wouldn't have attracted the other people to our community of practice around FOSS. These are new people who are more than 80% in agreement with free software practices. People who have brought so much innovation to the space around free and open source software that it is now a ubiquitous and important part of the human experience.

Bottom line - it's more potato/potahto than anything. I'm using the brand that has traction and impact outside of the technology sector, and we call that the open source way.

== S12 ==

The Open Source Way: Creating and nurturing communities of contributors

  It's a handbook you can rebrand and use to learn how to do and bolster usage of the open source way.
    http://www.TheOpenSourceWay.org/book

  It's a wiki you can participate in; help your community of practice document the knowledge on what does and doesn't work.
    http://www.TheOpenSourceWay.org/wiki

=== S12 notes ===

I am way stoked to be able to present this to you today. I'm here at Open Source Bridge, which in that name is branding itself as a connection between worlds that is done the open source way.

This book is about applying the open source way to just about any endeavor. It is deliberately beyond just-technology, taking the brand strength to business, education, government, civic life, and so forth. It is written in a direct style focusing on what to do, what not to

do, and how to do it. Each section is a paragraph explaining the principle (the what), a paragraph of implementation details (the how), and a paragraph of example (the why). Then on to the next section.

It is written by a community of experts; it is lightweight, about 30+ pages; and it is incomplete on purpose. One of the principles in the book says to leave room for participation. If there is not enough obviously to be done, people will feel it's all done and not stay to participate. If there is too much to do, it's scary to a new potential contributor. There is a balance.

Also, I wanted to see that we grew examples that weren't just me talking about Fedora again and again.

This book started as an internal project last year, what our team termed a cookbook. It would document the recipes to success that we found ourselves telling and teaching and repeating over and over. It would be lightweight and derivative, not trying to rewrite the world but to capture and distill. As it happens, that is part of the team's mission for Red Hat.

One of the lessons we are there to teach is that what is good for the community around FOSS not only grows well there but is also good for Red Hat. Even if it is just a splash, if it helps to rise the tide for all, Red Hat rises with that tide. So, it made immediate sense that we should use our positions as thought leaders to bring this distilled knowledge to the wider community of practictioners who can benefit from using and contributing. Like a barn raising, we did all the start work but made sure there was plenty for others to be part of the excitement.

Was it our Community Architecture team's way of giving back? I think it was as much that as a way of protecting against being eaten by raptors. It doesn't do us any good to be the only ones who can explain and get things done using these methods. We also know that as good as we are, we are only always going to be a few of us compared to the world. As with many other endeavors, it's clear that a free and open community content approach is going to yield better goods, over time. Not being driven by a commercial imperative, we don't have to acceed to publishing house NC requirements. If what we get over time is good enough, someone is going to want to make a dead tree version anyway, and there aren't any field of use restrictions, so it's free forever.

From a simple free content writer standpoint, it's an awesome opportunity. It's hard work to collaborate with people across time and culture zones, and you have to learn to give up individual control of all aspects of the process or product. But that isn't wildly different from what writers do anyway, and in a FOSS community, you can directly control your own fate.

So you'll see the website using my favorite tools, which earned that favorite status from years of beating on them in the Fedora Project. Same thing with the freeing of content. The book is under the Creative Commons Attribution-Share Alike 3.0 Unported license. The tools used in production are MediaWiki and MySQL running on Red Hat Enterprise Linux 5, and Fedora Hosted for mailing list and the git repository that houses the DocBook XML. We build the DocBook using

the Publican tool that Fedora Documentation and Red Hat Content Services use, in the same way for the same kind of purpose.

We custom created an environment that I'm sure works for contributors, and put at it's center a fireball of content that is ripe for use, but really needs contributors.

== S13 ==

* How does this matter to you?

=== S13 notes ===

As a person who is always pointing out when people are applying good free and open source principles, it's immediately obvious to me what this book is good for. There is a page on the wiki called [[Great stories to tell]]. This is where we are gathering and cajoling people to add to, somewhere to hold all the really great stories that illustrate the open source way. These stories typically demonstrate multiple principles, and so are too good to waste on just being an example for one principle.

One of those stories is so close to what has happened to me that I'm tempted to adopt it, but attribution is required since it is coming under the CC BY SA. Michael Tiemann tells this story, he was in San Francisco's Exploratorium with his daughter, and they came across a concrete pendulum experiment. The pendulum is a column of concrete that is banded in a wide belt of steel and suspended from the ceiling by a single cable. Considering its size, it must weigh several hundred pounds.

Hanging from the steel sides are magnets attached to a string. You can just reach the string with a bit of slack and give it a tug from the column. Michael tells, his daughter gave it a tug, the magnet popped off being so small from such a large mass, and she said, "Daddy, it's broken. Let's go." Michael then asked to give it a try. Here in his words:

> I told her to watch as I tossed my magnet, got it to stick, and then applied such a small force to the string that it was more a thought than a tug. I waited and did it again. After several such efforts, the first motion became visible. I timed my impulses with the pendulum, and within 5-10 minutes the motion was so great that the cylinder's swing exceeded the length of my string and the magnet flew off. Resonance naturally amplifies even the smallest of properly coordinated incremental impulses.

That's from Michael's column on opensource.com, "Amplifying creativity and business performance with open source", published in February 2010.

(Did you notice that attribution? Free content, it's that simple.)

== S14 ==

[exploratorium pic]

=== S14 notes ===

I've been there with my daughters and I've done the same demonstration. I've seem them apply the same principles in other parts of life. So, to me, there is a clear connection between the wider nature of the world and the microcosm communities where these practices, this open source way, can positively influence.

How does this matter to you? I'd like to finish by tying together what I'm saying about the scientific support via communities of practice and implementing the open source way. I'll be broad but hopefully these spark an idea from you or a neighbor. Or ask a question when I'm done.

1. You have or work at a business where a significant portion of your server software is running entirely on open source. You may personally run a Linux desktop, even use it as a primary system for work. You can clearly see that free and open source software could take are of all of your software needs.

There are a number of ways a clear handbook can help you. It's possibly valuable to point at the Red Hat branded version, the talk about scientific methods, and the compelling arguments.

2. But your position is really close to that of the business needing to gain that extra advantage of open source, the one that is beyond no-cost. This is where the real value is. Instead of being held prisoner to the feature churn, you help make happen the parts that are important to your business. I encourage you to take a look at a short video featuring Michael Tiemann called "The Open Source Triple Play". The basic idea is that by taking control of the tools that produce the solution you need, you get more of what you need, built better, and faster. The people supplying the tools you start with don't have to waste resources iterating with you when you are better suited at deciding what should be done where.

3. Do you develop software or websites that run on free and open source software? It's probable you already participate in communities of practice. A handbook like this helps you create leadership and learning experiences that are valuable to you and your fellow community members.

4. Are you a student or an educator looking for a learning-rich environment that allows you to make your own way, build your CV, learn constantly, and have a good time while making a difference? Participating in open source projects can be like an internship in terms of the experiences, skills learned, and interpersonal networks created and strengthened. Having a handbook is one way to get yourself ready for participating, and it gives you a method to find ways to help in implementing the open source way across a project.

Remember, the open source way is more than just putting a license on some code or content.

Listening to Leigh Honeywell's keynote this morning, I noticed how many of her principles have a corrolary in the communities of practice, and how many tied directly to The Open Source Way.

I hear of many cases of people trying to spread the story of how the open source way can inform their domain. It could be church, a gardening club, a university, or a business with a non-technical core competency. In all these cases, a book like this gives the chance to contribute *your* example, your story. It is immediately applicable for the full feedback loop that is the open source way embodied in a book.

== S15 ==

* Questions
* Links
  http://www.TheOpenSourceWay.org/wiki
  http://www.TheOpenSourceWay.org/book

=== S15 notes ===

##

== S1 ==

*

=== S1 notes ===