Why do we do, the things we do?
Sha na na na na na.

Oh, and open source business models.

...title...

Lately, I've been teaching my daughter pig latin.

Actually, a family variant called 'Arb'.
Parbig Larbatarbin!

My father and uncle taught it to me when I was nine. Like me, she is tickled to be able to say things that sound really silly that her mother can't understand. Silliness is important in our family. Family humour is a very specific thing. Some families like teasing.

Narbo warbay marbom!
Other love pranks and practical jokes (not mine, thank goodness). On my father's side, the family loves puns.

On my mother's side, the taste runs to jokes and riddles, and preferably absurdist ones. As a young boy, I liked the ones my mother collected, simple ones from joke books and the like.
What’s purple and glows?

What's purple and glows? An electric grape!

But as I grew older, my mind craved more twisted bits of language. I still remember the moment, I must have been around 10, when my uncle dropped this gem at the dinner table.
Why does a duck?

Because one of its legs are both the same length.

Let that jiggle around in your brain for a while, like a mutant worm.
One of its legs are both the same length.
In university I was floored by the #1 item in Letterman's "top ten stupid questions we have been sent", which was

How many raisins in a dollar?
So, I'm a practitioner and admirer of the absurd,
And so,
I'm pleased to be in the land that gave us, back in the 40s
"Military intelligence",
And more recently,

"Keep your government hands off my Medicare."
America has a rich tradition of absurdity,
And I assume it's as a practitioner of the absurd
that I've been asked today to talk about
Open source business models

Open source business models
<sarcasm>

hmmmm
Because we're all in open source
for the money,

right?
open source

- community
- shared accomplishments
- satisfaction of learning and exploring
- freedom to build
- membership in a global community

The sense of community, the <X> shared accomplishments, the <X> satisfaction of understanding how something works all the way down, the <X> freedom to build exciting new things without asking permission first, the <X> chance to work with a global community of talented and like-minded individuals.

Those are all secondary considerations.
The important thing is,
and it's surprising how often this
"objection" to open source is raised,
The important thing is,

If ESRI gave away their software for free, they would go out of business!!! (ack!)

how can a traditional software company make money
selling open source software?
The logic is straightforward,
ESRI exists because their software costs money.

Your software does not cost money.

if the existing software companies make money by walling off their intellectual property and selling limited, licensed access to it, it necessarily follows that

if a business does *not* wall off their intellectual property
Therefore.

and sell limited licensed access to it

You do not exist.

Q.E.D.

then the business will not survive.

Q.E.D.
The 451 Group, an IT industry analysis company, surveyed 114 “open source companies” and they came up with the best explanation of open source business models I have seen so far:
“Open source is not a business model.”

“Open source is a business tactic, not a business model.”
Open source is not a market in and of itself,

“Open source is not a market in and of itself...”

... nor is it a vertical segment of the market.

“... nor is it a vertical segment of the market.”

nor is it a vertical segment of the market.
Open source is a software development and/or distribution model that is enabled by a licensing tactic.

OK, so what is a business model?
"A business model describes the rationale of how an organization creates, delivers, and captures value."

A business model is a process that bundles something of value. Where do we create our value?

Business models can be as simple as building an item from parts and
more value

selling the higher value result.

But they don't have to be.

value

The most disruptive models tend to be the most surprising ones, because they invert our notions of where the intrinsic value of a product lies.
The initial business model for newspapers and newsmagazines was very simple.

The <X> publisher assembled the newspaper from original writing, <X> laid it out and printed it. <X> Readers paid the publisher for that product. The <X> value was in *content* delivered to a *reader*.
In 1729, Benjamin Franklin started publishing the Pennsylvania Gazette, and he did something very odd. He sold each copy for a fraction of the production cost, so he was losing money on every copy. But he made up the difference via sales of advertising.

He was still <X> commissioning original writing, <X> laying it out, printing it and <X> distributing the product, but the <X> *value* was in the *readership* delivered to *advertisers*, not in the newspaper itself. The lesson of the newspaper business is, you don't have to sell a thing to capture value with it. For a more recent example of indirect value capture...
look at the Amazon Kindle 3G.

The parts cost <X> $156.
The device costs <X> $189.
That leaves just <X> $33 per item to cover all R&D, marketing and cell data charges.
Amazon is basically giving the Kindle away for free.

But the zero-profit Kindles are enabling a very high profit business in shipping digital books to customers.

So, it's not unsurprising if a business invests in an unprofitable secondary business to create value in a profitable primary one.
Or consider the venerable concession model.

Why does every museum have a gift shop?
Surely not for educational purposes?

Why does movie popcorn cost $7.50?
Why does an egg McMuffin cost $2 at McDonalds, but $5 on an airplane? These lucrative side businesses have exclusive access to a buying population that is primed and ready to consume their wares. So there's *value* in access to a market of pre-qualified customers who are ready to buy.

In their study of open source businesses, the 451 Group noted that were a large number of different tactics for creating value in the open source world, and that any given business might employ several of them simultaneously.

Here's some of the tactics commonly associated with extracting value around open source.
Dual licensing

Popularized by MySQL and Sleepycat software (both, curiously, now wards of Oracle corporation)
offers a GPL version of the software for free but

charges for access to a non-GPL version. The value is primarily for companies who want to embed the software for re-sale, but also derives from misunderstandings and paranoia around the GPL.
Open Core is a model practiced by relatively new, venture funded business like Xen, SugarCRM and Nagios.

Open core combines an open source core project with proprietary extra features to provide sale value. The sale value is in the extra features.
Dual licensing and open core require the company to hold all copyrights.

Both dual license and open core projects tend to be captive of a single company, since in order to carry out their dual licensing scheme the company must hold the copyrights to the open source project, which means they usually employ the core contributors directly.

What these models have in common is that they look a lot like proprietary software companies.

Open core, and dual licensing and proprietary licensing all capture value.
The value being sold is access to intellectual property.

in the same way, by charging for access to intellectual property.

This is perhaps why these models have been easiest to evangelize and have received the lion's share of venture funding over the past several years.

Moving away from the traditional software company model,

my favourite system is articulated by James Dixon of Pentaho, for what he calls "professional open source software".
Imagine a bee hive.
Now, imagine you want some honey.
Knowing nothing about bees,
you might be understandably leery
about just opening up the hive
and trying to scoop some out.

What you need, is a bee expert!
A bee keeper!
The beekeeper mediates our relationship with bees. He <X> provides the bees with hives and care, and the bees generate raw honeycomb, which <X> he processes, packages, markets and <X> sells to us for money. Which he uses to buy hives and so on.

Even though he doesn't make honey himself, the beekeeper *adds value* to the product we buy.

In Dixon's model, open source developers are the bees.

Developers, let's hear you all buzz. .......

Users, that's a pretty intimidating sound, right? Maybe you want someone who can work with the bees to help you get the best honey?
In this model, the "professional open source software" company is the beekeeper. They provide development support to the community, and they also package the raw software for use by customers, adding things like installers, documentation, uniform branding and messaging, and support agreements.

The professional open source company completes a virtuous circle, where the value in the raw software produced by an open source community is fused with the value from the company to become a salable product, which provides resources to employ staff and provide support that in turns strengthens the open source community to provide yet more valuable software.
Take apart a proprietary software company, you'll find development, product management, sales, support, documentation, training. The "professional open source company" is the result of looking at those parts and asking "is there still value here if we move most of the development and intellectual property outside the organization?"
And the answer is yes.

Dixon’s company, Pentaho, operates with this model. So does <X> Red Hat. So does the company I work for, <X> OpenGeo.
Professional open source value

- Unified marketing story
- Integration of diverse components
- Installers and ease-of-use
- Training and documentation
- Contractual levels of support

Professional marketing materials provide value to purchasers looking to rationalize their decisions.

<X> Integration provides value.
<X> Installation provides value.
<X> Training and documentation provide value.

But perhaps the biggest value sold by the professional open source companies is the <X> support contract.
We say we're providing "support", but what we're really providing is insurance

What, me worry?

for when things go wrong.
Real insurance companies protect against disaster by
hoarding a large pile of money to fix things afterwards. Professional open source companies protect against disaster by hoarding a large pile of intellectual capital, in the form of ... not disembodied brains ... but core developers in their chosen projects.
So Red Hat employs Alan Cox, the Linux kernel guru, and Pentaho employs Matt Casters of the Kettle project, and OpenGeo employs, well, me and many others.

There's a whole other category of open source business that employs core developers, too, the product specialist.

A product specialist company also employs core developers, but they frequently get paid for non-core development tasks. However, core developers on staff allows them to add new features for clients, and to perform complex system integrations with the open source code.

But wait, aren't I just describing a consultancy? The goal of the product specialist is to become not just *a* consultancy, but *the* consultancy.
To dominate the mindshare around services for a particular project. To be the biggest fish in their pond.

So
<X> MapGears tries to be the go-to company for MapServer, and
<X> CampToCamp tries to be the go-to company for MapFish, and
<X> Oslandia tries to be the go-to company for PostGIS, and
<X> RadiantBlue tries to be the go-to company for OSSIM and
<X> GeoCat tries to be the go-to company for GeoNetwork.
Product Specialist

1. Invest R&D effort in building open source software.
2. Gain useful experience and a reputation for expertise with the software.
3. ...
4. Profit!

Product specialists usually spend less money on sales and marketing, they spend their spare resources on core development, to further hone their reputations as the top experts in their field.

Folks back home ask me how I get invited to speak at all these conferences, and I tell them it's because I'm world famous, as long as you ask the right very small sub-set of people.

The same thing applies to product specialists, the niche they aim to dominate may be quite small, but it's still big enough, on a global basis, to support a profitable business.
So far examples have all looked a lot like existing software companies, but there is lots more value available in the world of open source!

Any open source project, if sufficiently popular, probably has a side market for accessories of all kinds things that aren't the software, but add value to the software, that make it prettier.

ac·ces·so·ry
/akˈses(ə)rē/

Noun: A thing that can be added to something else in order to make it more useful, versatile, or attractive.
Production-oriented services

Production-oriented services and add-ons are a form of accessory.

In the Linux world, the Red Hat Network is an example. For JBoss, the JBoss Operations Network provides tools that make it easier to operate the software in data centers.
Deployment platforms

Deployment platforms are an accessory that adds value by making it easy to put open source software stacks into production. Heroku provides this for Ruby on Rails, PHP Fog for PHP.

Accessories range from these kinds of
highly technical services at one end, all the way down to plush toys and CDROMs at the other end.

In general, there is value to be had in selling "accessories" to popular open source projects.

These businesses don't necessarily look like traditional software vendors anymore, because they don't necessarily spend much of their time working on the core, and they extract no value directly from the core software.

ac·ces·so·ry
/ak'ses(ə)rē/

Noun: A thing that can be added to something else in order to make it more useful, versatile, or attractive.
Are these "open source companies"?
Well, they are creating salable value through their association with open source!

So, there are all kinds of models for finding value in and around open source, and where are all sorts of open source companies.
“open source company”?

What about LinkSys?
are they an “open source company”?
They make routers,
but they put Linux on them!
They derive *value*
through not having to
license a proprietary OS and
by sharing development effort on
wireless chip drivers.

Or any of the many handset makers
using Android?
Are they open source companies?
They are definitely deriving
value through participating
in the Android community and process

Which brings me to the largest open source company in the world.
It's not Sun.
It's not IBM.
It's not even Red Hat.
<X> It's Google, the big G.

Measuring *just* volume of code released,

14 million lines of code

the 14M lines in only Chrome, GWT and Android put Google at the head of the pack in terms of global contributions.

But it is the fact that Google's core infrastructure,
miles and miles of racks of generic boxes running Linux would not have been possible but for open source, and the fact that Google engineers are active in the communities of open source projects Google depends on that really make the case for Google as the worlds biggest open source company.

But, notably, the place Google creates value has nothing to do with open source, Google creates value through INFORMATION ORGANIZATION.

where's Google's value?
In the words of their mission statement,

Organizing all the worlds data. Making it universally accessible.

Google's mission is to organize the world's information and make it universally accessible and useful.

But that information organization would not have been possible without open source.

If, in the early days, Google had to write or license all of the code they needed, they would never have gotten off the ground. <X> Without open source, the only organizations capable of running at web scale would have been those who already owned the necessary intellectual property in operating systems and file systems and databases.
Microsoft, Sun, IBM.

Imagine what the web would be, run by

Microsoft, Sun, and IBM.

But we didn't get that, instead we got a world
Only the strong survive. informed analytical organized

where success is determined by the ability to quickly gather, manipulate and analyze information.

Which is why librarians are in such high demand!
Oh wait, not librarians,
Marc Andreesen, one of the founders of Netscape, and all around smarmy bastard, wrote in the Wall Street Journal last month that "software is eating the world".

In support of his thesis, he points to numerous success stories, companies leveraging their superiority in software into superiority in the overall marketplace.
Borders being beaten by Amazon,

Blockbuster put out of business by Netflix
Disney having to buy Pixar

Kodak going under while Flickr flies
Recruiting

The recruiting market being taken over by LinkedIn

Telecom

The telecom industry being taken down by Skype and even the worldly success of traditional organizations
like Fedex and Wal*Mart

can be all be traced back
to superior use of information
technology and data management

The best organizations
know how to
extract value
from their
information streams.

These examples show
that victory is going to
the organizations that can most nimbly
extract value from their data streams.

And how do we extract
value from data streams?
software

With software!

And who creates software?

programmers

Programmers.

And what tools are programmers most effective with?
transparent
dependent
flexible
re-purposable

Tools where the answer is an e-mail or
Google search or
IRC question away.

Transparent, flexible, re-purposable tools.
Sure, I’ll send you a copy of the program...

Tools where there are no deployment limitations.

I’ll fire up 200 more instances!!

Tools where scaling is just a matter of spinning up instances.
Tools where the bugs can be caught and squashed in internet time.

I can’t believe Frank closed that ticket already!!!
Open source tools. Open source tools help you and your developers unlock value in information.

It's intellectually entertaining to look at the models of companies like OpenGeo and compare us to established proprietary companies and wonder if we're going to make it. Will our open source business model work? Who knows? *I* think so...

But it's not that relevant to *you*. 
find your
own
open source
business model

You need to have your *own*
open source business model.
You need to do the research
and understood where open source
can add value to your business,
to your agency, to your department,
and I hope that's why you're
all here.

This week, go to the sessions,
talk to the developers,
talk to your peers,

join the open source community,
and find your
open source business model.

It's out there waiting for you.
tharbank yarbou