

Thank you all for having me here today.

Given the title for today's talk, I hope you will not be surprised to find that the body of the talk bears little relation to the headline, the important thing here is that you clicked on the link.



So, I'm not a CEO, or a CTO, or a VP, or a Deputy Ministry or Executive Director of anything, at least I haven't been for a good 10 years, so if you're wondering why I'm here, that's totally fair.



Hi, my name is Paul.



British Columbia

Consultant

Needs Analysis

Requirements

Documentation

• RFP

• RFI

RFQ

Developer

Oracle

ArcView

ArcMap

Intergraph

Geomedia

Microstation

Esri

I'm a person who knows geospatial software pretty intimately, as a developer of open source GIS software, and a senior employee of a open source support company. But for this group, most relevantly in my former life as a consultant to government,

where I worked with all the proprietary tools, and worked through all the organizational processes.

"yea, I have walked through the valley of the shadow of death, and I fear no evil"

I have lived the dream of doing software and geospatial in large organizations.

And, after decade and a half of that, I decided that devoting myself to open source would be more fun.



But before I get to the cool open source stuff, I'd like to start ... with a digression.



One of my favourite pieces of poetry was delivered, not by a beat poet in Greenwich Village or by a 19th Century Romantic,



but by one Donald Rumsfeld, then Secretary of Defence, from the Pentagon press briefing room, on February 12, 2002.



Hart Seeley later formed the Secretary's words into a poem which was published in Harper's Magazine in 2003 as "The Unknown" As we know, There are known knowns. There are things we know we know We also know There are known unknowns. That is to say We know there are some things We do not know. But there are also unknown unknown The ones we don't know We don't know. ... read ...

I've also found the same sentiments expressed less elegantly, but more forcefully in diagram form, showing

The stuff we know we know;





The stuff we know we don't know;



And the vast expanse of stuff we don't know we don't know. It is scary that the largest category by far is one we definitionally cannot comprehend, the stuff we don't know we don't know.

Of course, this is an epistemological diagram of all knowledge,

so we can **constrain** it, a bit, by noting that,

for practical purposes,

we are really only concerned with the



stuff we should know.But unfortunately the stuff weshould know still falls in all three categories.As professional knowledge workers, one of our jobs is ...



to constantly **expand the size** of those green and yellow circles, to **grow the scope** of our knowledge **and** also grow our awareness of our **limitations**.



It's why we come to gatherings like this one. We gather around the warmth of the ... overhead projector. And we tell each other stories.

I'm going to tell you a couple stories this morning, about open source and open data, but I'd like to preface them by noting two larger contexts that might colour how you receive them:



The first context, particularly if you are in government, but also in most larger corporations, is your professional life is probably lived within a heirarchical organization.

You have a boss, and she has a boss.

When you go out into the world, it is as much as an agent of your organization as an individual, you identify as "a **somerole** from **someorganization**" rather than as "**someone**".



The stories I am going to tell, about open source software and open data, are about knowledge work that is fundamentally non-heirarchical, and that may immediately disqualify them in your eyes. It may signal of a lack of seriousness, or authority, and it might make you take a mental step back. Notice that moment, see if you can resist it.



The second context, as members of the GIS community, it's hard to overstate the extent to which the full-spectrum dominance of one corporation colours our understanding of what is possible. From education to professional development to the stories we tell each other at these events, our understanding of "how to do the job" is, understandably, coloured by the tools at hand. When all those tools come from one company, and all our stories come

from one company, it's not surprising that we frequently find that "we do not know what we do not know".



Anyhow, my first Story... Once upon a time, Once upon a time, there was a young man with wild ideas about freedom,



who took on the established order of things, appeared to lose, but in the end changed the world forever (though perhaps in ways he might not approve of).

Actually, not that young man,



though there is a striking resemblance...

In 1980, Richard Stallman was a programmer

	at the MIT Artificial Intelligence lab. Some of the best minds in the
MIT artificial intelligence lab circa 1980	

the best minds in Al

artificial intelligence field worked together

and shared ideas and implementations of those ideas in code.

It was, to hear Stallman tell it,

golden age
of hacker collaborationa brief golden age of
collaboration and intellectual ferment.Then one day,
and don't all horror stories start this way,
one day,



sharing ideas and code

the lab got a new printer (a xerox 9700). Unlike the printer it was replacing, the new printer came with a binary-only printer driver; the source code was not included.

Stallman had modified the previous driver to send a message to users

when the printer jammed. With the new binary driver he couldn't do that. The situation was *inconvenient*, it was a pain. printer has jammed Why couldn't Xerox just share their code? Everyone would be happier! Most people might have shrugged. why not just share But for Stallman it was a the code? galvanizing moment. Over the past five years working in the AI lab, he had grown used to sharing code and ideas with other programmers. But now the atmosphere in computing was changing. It wasn't just the printer driver. nings were changing...



Did you know that: symbolics.com was the first .com domain name? A private corporation had started recruiting his colleagues in the lab. Once hired, they were no longer allowed to exchange code with him.

The old computers in the lab, and the software that ran on it, were becoming obsolete.



The fancy new VAX computers being purchased by the lab included operating systems that were locked down: you had to sign nondisclosures just to use them.

It was the death of the old collaborative community. Stallman worried that



"the first step in using a computer was to promise *not* to help your neighbour" by accepting a license agreement.

As a highly talented and idealistic computer programmer, Stallman wanted his work to serve a larger purpose.

Facing the death of his old intellectual community, Stallman asked himself



"was there a program or programs that I could write, so as to make a *community possible* once again?"

You can't use a computer without an operating system. So Stallman decided that, first he needed to write an



operating system. It had to be <X> portable to many computer platforms, it should be <X> compatible with the popular new UNIX operating system, and most importantly it <X> should be **free**.



By "free", Stallman meant you should be <X> free to run it, you should be <X> free to modify it, you should be <X> free to share it, you should be <X> free to share your modifications



In a latinate language like French, Spanish or Italian it's more obvious that, Stallman isn't talking about **price**, not "logiciel gratuis", he's talking about <X> logiciel libre, <X> software libre, <X> il software libero. He's talking about



liberated software... The key addition is LIBERTY.

So, in 1984,

rather than join a computing industry that he considered morally bankrupt, Stallman decided to basically start a **new one** from scratch.



It was an audacious plan. Stallman called his new system GNU, which stands (recursively) for

GNU's Not UNIX

"GNU's Not UNIX?" (See the recursion?) What's GNU?

GNU'S Not UNIX What's GNU?

Not UNIX Not UNIX Not UNIX GNU's Not UNIX



Let me just take a very minor diversion here to add some extra flavor.



In order to ensure GNU remained free, and did not get subsumed into a proprietary system in the future, Stallman released his work using a scheme he called "copyleft".

Generally speaking, intellectual works (books, movies, songs, computer programs) are either under

©	public domain
all rights reserved	no rights retained

copyright or public domain. The author either retains full control over the work, "all rights are reserved", or no control, "no rights are retained".

Copyleft, and open source licenses in general,



use the copyright system to selectively grant permission and exert control over software through **licensing**.

Authors retain copyright, but grant liberal usage rights via a license.

The copyleft license grants permission to all recipients of the code to use, modify and redistribute

The license requires that any redistribution of the work or derived products include the source code, and be subject to the same license. The legal language can get complex, but the principles are hardly foreign.

Version 2, June 1991 Copyright (C) 1989, 1991 Free Software Foundation, Inc.,

GNU GENERAL PUBLIC LICENSE

51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Lesser General Public License instead.) You can apply it to your programs, too.

> Share and share alike. Do unto others as you would have them do unto you.

to guarantee your freedom to share and change free software



OK, back on the highway.

So, in 1984 Stallman quits his job at MIT and starts working on GNU full time. No visible means of support, this is a labor of love.



But where to start? From a blank canvas, you want a completely free software ecosystem, what do you do first?

If you wanted to build a 100% all hand crafted house,



you would start by hand crafting your tools. Stallman did the **same** thing, with GNU versions of software development tools.

He starts by writing a text editor



(GNU Emacs), so he can write his free system using only free tools.

The Emacs editor proves so popular (and internet access is still so rare)



that he is able to earn a small living selling tape copies of the code (distributed under copyleft of course).



Then he writes a compiler, GCC, You can still find GCC in every Linux distribution and also in Mac OSX.

Stallman lives like a monk, works like a demon, attracts some followers and helpers,



who formalize the project in a foundation.

By 1990 they have most of the components of an operating system.



Most importantly, they have a full programming tool-chain: compilers, debuggers, editors, core libraries, and so on. All the things you need to write complex software. What they don't have, is a UNIX kernel, the piece of software that talks directly to the hardware.



At this point, all their free tools are still being run on proprietary UNIX!



ОК...

In 1991, a Finnish computer science student



named Linus Torvalds, uses Christmas money from his grandmother to buy a new computer, an Intel 386.

As a student at the university, he has access to UNIX systems, and he wants to run UNIX on his 386 at home.

This is not possible.



The good implementations for the 386 cost more than the computer itself. The cheap implementation, Minix, is quite limited.

So Linus writes his own kernel.



He uses Stallman's GNU tools to write and compile it. And in August of 1991 he posts the following on an internet discussion list.

Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). Hello everybody out there using minix -

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons)

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi)

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-) Linus (<u>torvalds@kruuna.helsinki.fi</u>)	Underneath the technical language, note the subtextual bits: the humility (just a hobby), the interest in other people's ideas (what do you like/dislike in minix).
does anyone want to play?	The posting is an invitation. Does anyone else want to come out and play? Does anyone? They do. Within 15 minutes, he has a reply.

Tell us more! Does it need a MMU? (memory management unit) How much of it is in C?"

"Tell us more! Does it need a MMU? How much of it is in C?"

Within 24 hours,



he has replies from Finland, Austria, Maryland, and England.

In a month the code is on a public FTP server. Within four months, it is so popular that an F.A.Q. document has been written to handle the common questions.

Linus Torvalds tapped a seam of enthusiasm just dying to express itself.



People who loved computers and computing and just wanted to play together. And through the medium of the internet, using only the simplest computing tools (diff, patch, ftp, e-mail), he built a community of thousands of contributors, and



together they built a usable operating system.

Something important changed between the time Stallman started the GNU project and when Torvalds released Linux.



The values of collaboration were the same, but the opportunity to exercise those values was greater, via the internet.

<X> When Stallman started GNU in 1984, there were 1000 hosts on the internet. <X> When Torvalds started Linux in 1991, there were over 400,000. And the pool of potential collaborators was in the middle of a huge expansion.



Permit me one more short digression on the digression to talk about



starwarsuncut.com

and talk about Star Wars. In particular, let's look at a web site called Star Wars uncut.



Star Wars Uncut has taken the original movie and chopped it into 473 fifteen second scenes. Each scene is then separately claimed and re-enacted by site members, and uploaded. The result looks like this.

frivolous but why now?

Seems pretty frivolous, right, but break it down. How is this (frivolous) collaboration possible? And, why is it only happening now, <X> not 10 years ago? There were just as many Star Wars nerds 10 years ago as there are now.



First, this activity requires easy access to video recording and editing tools, and until recently cameras and video editors were very expensive. <X> And it requires enough bandwidth to download and upload video, and until recently people didn't have that kind of bandwidth in their homes. <X> And finally it requires Star Wars geeks.



To build a large collaborative product, you need <X> tools freely (or very cheaply) available and you need <X> sufficient connectivity between participants. Combine that basic infrastructure, with community, collaboration and <X> love for the subject matter, and magic happens.

There are many, many more examples of this kind of group collaboration,

the academics call these collaborations "commons-based peer production".

"commons-based peer production"

open source is "commons-based peer production" Open source software in general and the Linux project in particular is one of the earliest examples of internet-mediated commons-based peer production.



You may have heard of a little project called Wikipedia. It's a classic example of commons-based peer production. People contribute the little bits of knowledge they have, and together build up an edifice of knowledge that is more complete than anything a hierarchical organization ever could.



It's not uncommon for open projects to have some kind of a action/ reaction origin story. Linux was a reaction to the limitations of Minix and cost of Unix.



Wikipedia was founded in reaction to the historical encyclopedia companies,

who controlled who could contribute,

and locked out contributors who had plenty of knowledge but lacked formal qualifications.



From the founding of Wikipedia in 2001, there was an explosion of editors, people who wanted to contribute but had previously been held back. As the content has filled in, the number of active editors seems to be converging to about 40,000.



My next story also has an action/reaction format, and it's also the story of a common-based peer production project springing up and transforming an industry.



Once upon a time, on a far away rainy island, throughout the 1700s and 1800s, the English wrestled with a number of difficult administrative problems:

subduing the Scots





taxing the Irish



and not being invaded by the French In all these tasks, they found that having accurate and detailed maps of the territory in question was really useful,



so useful, that they established a quasi-military national agency, the Ordnance Survey, to handle all the mapping.

What with all the wars, and the colonizing, and so on, Ordnance Survey got in the habit of mapping the UK to a uniform standard and at a very fine scale:



by the end of the 19th century, over 400 towns were mapped at a scale of 1:500. That's right, 1:500, five metres to the centimetre. So, nation-wide mapping effort, but detailed enough to capture phone booths and mail boxes.



This is where things get interesting, at least from the point of view of decentralized polities like Canada and the USA: Ordnance Survey mapping was so good that, by the time the cities and counties got around to wanting maps of their own, they were happy to just use Ordnance Survey data as their base, and they were willing to pay for it.



So from an early time, Ordnance Survey became the monopoly source of first maps and then later on the monopoly source of digital map **DATA** in the UK.



Now, fast forward to the 21st century: if you want to make a computer map in the UK, you first buy some Ordnance Survey data. That's the way it works, no exceptions, and Ordnance Survey data is so good, so detailed, so accurate, that the barrier to entry for any other provider is very high; they sit on a natural monopoly, with all the concern for customer satisfaction and user experience that implies.

(ie, basically none)



And what if, in 2004, you're interested in building up a map of bicycle routes in the UK, how do you get a base map for that? The answer,

if you don't have some kind of organizational affiliation and access to the Ordnance Survey base map is: you don't. You're out of luck.



Except in this case, in 2004, a bloke named Steve Coast said "to hell with that", and decided that a crappy map he **compiled himself** was better than no map at all. And he took inspiration from the Wikipedia project, and instead of just compiling the data once, for himself, he built a simple system to share his map,

and allowed other people to add to it and edit it.



What happened next was an explosion, as OpenStreetMap went from a UK project, to a European project, to a global project, with coverage everywhere, and a mass usage. The patterns in the historical edits are amazing: the initial burst of light as the empty spaces are filled in, with local mapping and data imports; then the slow burn as imported data are cleaned up and modernized; and then more flashes as things like routing information and addressing information are brought in.



OpenStreetMap has been getting more and more detailed, both in terms of the features it maps, and also in terms of the kind of information **on** the features.

Five years ago, OpenStreetMap was completely unsuitable for routing; **now** there are companies built on the idea of routing over OpenStreetMap data.

Five years ago geocoding was spotty on OpenStreetMap;

now there are whole countries with complete address registers in OpenStreetMap.



Now, I'm guessing there's a lot of professional data maintainers in this crowd...

It's easier to hear that the internet has disintermediated travel agents, or that open source had commoditized software, or that wikipedia has immolated the encyclopedia industry, but when the internet juggernaut starts to take over data creation... that hits a little close to home.



"Would you trust these people to make your maps?" And there's some common responses, which I've seen in the software field as well.

First, folks look at the grass roots of the initiative and say things like



"that's just a bunch of hobbyists, would you really entrust your big, super-important business to something put out by a bunch of basement dwelling nerds?"



"Would you trust these people to make your maps software?" Well, open source software is a commons based peer product that has been around longer than OpenStreetMap, so we can kind of see history in action, and for open source the answer to the question is an unqualified "yes".



Your Android and iOS smart phones run open source operating systems.



Your Google searches are executed on servers running open source.



Your Facebook updates are stored in open source databases, and run on open source languages.



Your home router is running open source.



So is your thermometer,



Your crockpot,



And the seatback movie system I watched on the way here.



By lines of code released, Google is the largest open source development shop on the planet. Facebook is second. Companies as diverse as Netflix and Amazon not only build infrastructure on open source, but also release their software as open source.



In 2001, the then-CEO of Microsoft, Steve Balmer compared Linux to **cancer.** Also communism. But seriously, **cancer.** Now Microsoft Azure is dominated by Linux. Open source is the new reality, and even a company like Microsoft, founded on proprietary software, can only bow to the inevitable.

https://www.zdnet.com/article/linux-now-dominates-azure/



Nor is open source adoption limited to the private sector. Big government organizations build infrastructure on open source.

O National Geospatial-Intel ×			Paul	*
C GitHub, Inc. [US] https://github.com/ngaged	int		\$	=
O Seen Othe	Explore Gist Blog Help	pramsey +-	4 ¢ ₽	
National Geospa	atial-Intelligence			
Agency o		People	1>	
A nueso press co usbenismen uderum		chrisbennight Chris Bennight		
Filters + Q, Find a repository				
disconnected-content-explorer-example	S JavaScript *0 // 2			
Example reports for the Disconnected Interactive Content Exp	sioner (DICE) app			
updeted 14 minutes ago				
geowave	Jeva ★ 52 🖓 12			
GeoWave provides geospatial and temporal indexing on top of	I Accumulo			
Updated an hour ago				

The NGA, the spy satellite guys, even have their own GitHub account.



- **Trend** is upwards to more use
- Some major institutions have adopted



 Most major institutions have adopted What I started talking about open source to GIS folks, five years ago, I would note that open source was an unstoppable trend, and that the time to start learning is now. Now I can only note that open source is the dominant reality, and the time to start learning was yesterday.



For the private sector, OSM is already a dominant map data reality. As with open source, for the public sector, the trend will be to follow behind, and that trend is already started.



The second response folks have to collaborative knowledge projects, like open source and open street map is the appeal to authority: well, that data may be pretty and all, but it's "not authoritative", it could have come from anywhere, the data is un-hygienic, you don't know where it's been.



But here's the thing: just like rock beats scissors, up-to-date and comprehensive beats authoritative.



Back in 2013, after couple **hundred years** of publishing, the Encyclopedia Britannica announced that they wouldn't be printing encyclopedias anymore.



Not because they'd been put out of business by encyclopedia theft, or encyclopedia copying, or anything like that.

They've just been outcompeted. And not by the World Book, or National Geographic.



By a decentralized community of writers working together to build an intellectual commons around factual knowledge. This community is many things, but it is not authoritative.



Wikipedia is just another online intellectual commons. And it's not like Britannica didn't see it coming, they just couldn't stop it.

And, here's the thing about OpenStreetMap: it's already everywhere.



The maps in Snapchat? Those are OpenStreetMap.



Strava? Also OpenStreetMap.



In Tableau? Yep, OpenStreetMap.



The Financial Times? OpenStreetMap. And there's no doubt that OpenStreetMap pushes through changes quickly, that it's got the pulse of change handled.



When a Minneapolis bridge collapsed, Open Street Map, and thus all associated routing, was updated within the hour.



When a new \$2B highway was opened up in Vancouver a few years ago, Open Street Map had the new routes. So did Google. Apple, and Microsoft? Not so much... I note this because it was rather inconvenient for me when I tried to drive it. So, quick recap:



the academics call these collaborations "commons-based peer production".



basically communities of interest come together and engage in networked knowledge work and generate intellectual artefacts that would, in previous times, require very large hierarchical organizations to create

powered by collaboration	it used to require a major corporation like IBM or Microsoft to create an operating system, but Linux is built by a collaborative community
--------------------------	--



it used to require a major publisher like Brittanica to compile an encyclopedia, but Wikipedia is built by a collaborative community



it used to require a major government institution like Ordnance Survey, or Natural Resources Canada, to build and distribute national base maps, but OpenStreetMap is built by a collaborative community



And... That's all interesting? yes? But from the point of view of a GIS manager, say, so what? Does any of this move the needle on the day-to-day?



There's a couple perspectives to take to this. The first perspective is the passive one: from a traditional customer/vendor frame, there's a some active communities generating "product" in our field and maybe we should be looking at how that product can be used in our day-to-day.



So, should you be using the product of the open source software community? Yes, you should! And this goes for both staff and managers.



As a practitioner, as an employee in the geospatial field, open source is going to increase your ability to solve problems, the portability of your solutions, and your value on the market.

all the tools you need

- database
- desktop
- statistical analysis
- geoprocessing
- web services
 web browser APIs

In terms of ability to solve problems, the open source ecosystem includes all the technology pieces you need to run a geospatial office: database, desktop, statistical analysis, processing, web services, browser technology.

Open source databasePestGIS Industry standard OpenGIS standard

On the database side, the PostGIS/PostgreSQL database has become the industry standard. As an author of PostGIS, I am admittedly biased, but I think that you do most of the things we use GIS software for inside a spatial database using only SQL. And you might think I'm getting a bit grandiose to describe my little open source project as an "industry standard",



but that's basically the conclusion I drew when I noticed the Google BigQuery team citing PostGIS-compatibility as an important feature of their new support for spatial types. I'll be giving a talk later today about spatial SQL and PostGIS if you're interested in learning more.



On the desktop, QGIS is now a fully functional desktop system that any GIS analyst can get comfortable with in a few hours.



Without getting into a big demo, because as with any desktop GIS, there is an awful lot there,

Browser Corr Corr	
Raster	
Contraction and the second seco	sktop

all the data exploration tools for multiple data sources, database, file formats, tile repositories, and web services.



Analytical chains with a graphical analysis builder, including vector and raster algorithms, not just GIS but also image processing.



a huge number of cartographic options, paper production and map book functionality,



a rich collection of community plugins available for easy download, and the ability to write your **own** extensions and automations with Python as the default language.



All backed by an active community and online resources to teach yourself if you need to.

There was a time in my open source career when I still yearned for some of the features of ArcMap, but those days are long past, I haven't cracked it open in over five years.



The analysis and processing side of open source is built on the modular nature of open source development. Foundational libraries provide core capabilities, and those capabilities are used by applications like QGIS, but are also exposed in scripting languages like Python or R.

open s	ource analysis
GEOS	Computational geometry library
PROJ	Coordinate reference system conversion
OGR / GDAL	Vector format conversion and ETL Raster format conversion and ETL

The open source ecosystem is built on top of some foundational libraries,

the GEOS library for computational geometry,

the GDAL library for image processing and format support, and the Proj library for coordinate system reprojection.

These libraries can all be brought together into

the Python environment for building lightweight automations using a couple Python packages:



Shapely brings together GEOS and GDAL and Proj to support vector data processing in Python

Rasterio brings the GDAL library in to support raster data processing To me the best part about geoprocessing in Python

is not the geoprocessing part at all:

it's the access to all the other things that are available

in the Python environment, in particular the data science modules.



SciKit.learn is an amazing tool for building predictive models, Baysian models, linear and non-linear models, decision trees, that kind of thing.

If you're working with neural networks, PyTorch has become an defacto standard for configuring multi-layer neural networks.

And just to bring it back around to geospatial, there's a full geospatial statistics package, the PySAL spatial analysis library,

which does all kinds of spatial analysis models: auto-correlation, econometrics, markov models, and more.



And if Python isn't your thing, there are also bindings to the core open source spatial libraries for Perl and PHP and R and Ruby and Go.

One thing open source never lacks for (almost to a fault) is choice.



On the web services side, GeoServer allows data to be published from multiple kinds of databases, into multiple web services standards, both formal ones like the OGC WMS and WFS standards and informal ones like vector tiles and ZXY tiles.

It has a built in caching layer,

and supports dynamic caching so you don't need to pre-populate a tile cache.



And on the web client side,

components like Leaflet and OpenLayers and MapBoxGL provide a huge amount of functionality to build spatially aware applications,

and most importantly **don't require an API key** or tie your to a particular web services system, in the way that Google Maps API does, for example.



With all these tools at your fingertips, there really is no geospatial problem you cannot solve using 100% open source tooling. But you can also solve them using proprietary tooling, and maybe your office has a site license for something or other, so why bother with open source?



•

more value in the marketplace

That gets to the issue of portability. One aspect of portability is your ability to share your work with others. If you have built everything on a proprietary foundation, you've limited the audience for your work to just those people who share the same foundation.



There may be very worthy audiences you want to engage, but you can't, because you've chosen closed tooling. I love that PostGIS gets used by First Nations to store their geospatial data;



I love that QGIS is used to teach GIS in Africa; I love that OpenLayers is used by climate organizations to visualize impacts. I love that nobody had to ask me, **or anyone else**, for permission to do those things; they **just did them**.

be a stronger practioner:

 more tools to solve problems

more value in the marketplace

More selfishly,

learning tools represents an investment in your future career, and when you learn portable tools,

that investment can transfer into **any** new work environment.



When you devote yourself to Esri tools,

you are restricting yourself to work environments that are also standardized on Esri. Or worse, you end up deputized into the Esri sales force,

rolling into your new job with a **software shopping list** that must be met before you can start generating value. Far better to roll into your new job on the first day, and get down to business.

Job Description:

- data preparation
- spatial analysis
- cartographic presentation

Also selfishly,

do you know what people call someone who can take in dirty data in text files,

clean them up, and run a geospatial analysis in python, then put the results on a cartographic rendering on the web?



If she's lucky, they won't call her a "GIS analyst", they call her a "data scientist", because **then they'll pay her way way way more**.



One of the counter-intuitive side effects of Esri's systematization and commodification of geospatial analysis — well-intentioned work to make the field more accessible — has been a steady downward pressure on what a "GIS analyst" is worth on the marketplace.

http://mapbrief.com/



Brian Timoney is worth following on Twitter if only for his occasional highlighting of the kinds of things that organizations expect "GIS analysts" to do

	Brian Timoney Obriantimoney	~
The supp bit.ly 8:19 AM	latest euphemism in geospatial wage pression: "ArcGIS Server Analyst" //1cMCU53	
2 Likes		
Q 1	Cl 🗢 s 🖂	
		1

and the kind of money they expect to pay for those things. Just a quick survey of salaries

uið Alla	1 7 91	nalg 20	IGIIUSL
Salary Estimate		Salary Estimate	
\$55,000	(1304)	\$60,000	(25142
\$65,000	(1098)	\$80,000	(20262
\$75,000	(854)	\$95,000	(16086
\$85,000	(611)	\$115,000	(9962
\$100,000	(301)	\$130,000	(5667

on <u>indeed.com</u> serves up ... some estimates that make it pretty clear where the fruitful career path is.



Todd Barr wrote a great article a few years ago where he teased out the variables that would influence salary for a "GIS Analyst. Unsurprisingly experience is nicely correlated to salary, though a late-career analyst has over 20 years experience, and only reaps an average 30% premium for all that time.

https://hackernoon.com/gis-analyst-salary-breakdown-f9e6a1cfb2dd



Meanwhile, most skills have no affect on salary at all! If you want to get out of the analyst ghetto, the only two skills that offer any substantial boost are reporting and web development. Even better, **just stop calling yourself a GIS Analyst.**



For managers, the same principles apply, but in reverse. You want to **empower your employees** to solve problems, particularly to solve them without having to make requests for extra capital or operating dollars before they can start.



Open source software comes not just without licensing issues, but without license liability issues. The classic cycle of the "site license" is that you agree to a fixed site license fee, and then during the term your usage goes up and the next time you negotiate,

So that's some reasons for staff to branch out, what about managers?



your all "you can eat" site license fee turns into an "all that you ate" fee. We can kevtch about this being "fair" or "unfair" but it's undoubtedly a perfectly predictable consequence of building a business on intellectual property that **somebody else owns**.



It's always better to be the one with more leverage When you use someone else's intellectual property, and even more so when they are the only major vendor in the marketplace, your ability to dictate the terms of engagement takes a pretty predictable nosedive.

Who decides?	You	Vendor
How much this year?		X
How much next year?		×
Terms of use?		×
Development roadmap?		×
Interoperability?		X

Who decides: how much you'll pay this year; what you'll pay next year; what the terms of use are; what the development roadmap is; what software you're allowed to interoperate with. Even in markets with multiple vendors, switching costs alone can drive

Even in markets with multiple vendors, switching costs alone can drive organizations into dependent relationships with single vendors.



Giving your staff more value in the market place makes them happier staff, which leads to retention.

So, I've been talking about open source geospatial software to gatherings like this for quite a few years, and every once in a while, someone comes up to me and says

"that talk you gave last year totally changed my life" "that talk you gave last year totally changed my life".

Saying this about a software talk.

It is a totally absurd thing to hear about a software talk.

i'm not kidding

"that talk you gave last year totally changed my life" And yet, I have actually been told this several times.

The people saying it are technology staff in GIS departments, and the reason they say it is because adopting open source gave them a whole new toolbox to solve problems. some people think learning new things making new things without constraint **is really cool** The exhilaration of learning what was in that box, and the freedom to use that knowledge to make cool things, without external constraints (like licenses) on what they could make, was deeply empowering for them.

These are very special people,



they are the kind of people you want to hire.

I recently came across a diagram which explains it all in one page.



Take the personality traits of <X> intelligence, <X> obsession, and <X> social ineptitude.

People with intelligence and obsession are <X> geeks. Inept smart ones are <X> dweebs, and the inept obsessives are <X> dorks. Those with all three traits, in the middle, are the <X> nerds.

As GIS managers, building out new systems and pushing the envelope,



smart folks with a mapping technology obsession, geo-geeks ideally, but you can settle for geo-nerds.

So, how do you get those geeks and nerds to work for you? Offer something **interesting.** Remember, they are technology obsessives.



no, really. it can.

Whether you hire geogeeks or not, open source can actually help with staff retention. Really. This does work in the real world.



The city of Northglenn, Colorado wrote a report about their experience with open source, and they cited some of the motivations I've already talked about, but in the section on "Unobvious Motivations for Adoption" there is this quote: "Contrast an open-source implementation position with a 'defined skill set' job..."



"Contrast an open-source implementation position with a 'defined skill set' job..."





"...and the second is to call the vendor and wait in a telephone hold queue..."

"It is easy to understand why open-source jobs are prized."

— City of Northglenn, CO



"It is easy to understand why open-source jobs are prized."

There's a fair objection to open source adoption, which is, frankly, it is **HARD**. There is no sales team, there's no pre-sales team. Nobody will show up and scope your problem and tell you what components you should use in a solution.



I would like you to believe that THIS IS A FEATURE. THIS IS NOT A BUG. You and your team will have to download and try out the software you are thinking about. This will take time, time that could be spent doing other things.



But.

You will develop a **reality-based understanding** of the strengths and limitations of the software you are considering, you will not be trying to get by on a



marketing based understanding. And while I understand the appeal of a marketing-based understanding, in the long run, reality will serve you better.



So, I've been talking about why you, as users of geospatial software, might want to use and work on software that everybody owns in common, rather than software that some vendor owns and licenses back to you.

And there's sort of an unfair dichotomy there, between little innocent old YOU and the big bad VENDOR retaining control of their intellectual property.



And I'd like to take that dichotomy, and flip it around, and start talking about DATA a little bit, because, for at least those of you in government, there's a strong possibility that for DATA the role you play is that of the organization trying to retain **control** of its intellectual property.



And this is true whether you're maintaining a planimetric base map or a parcel map or even just a layer of city parks.



There's a tendency to want to get out the branding iron and stamp it "mine" and organize mentally around the idea that the **mission of your organization** is to "maintain and update and distribute this data".





"Our job is to make sure our citizens are well served by our data!" But I'd wonder if we can take a step back,

away from that mission, and see a larger mission.

Public sector organizations are there to provide services the citizens need.

And while those services sometimes reify into something concrete like "maintain our data",

for practical purposes the citizens rarely care who owns data they use, or where it's stored.

They care that it's correct, and that it's up to date.



There's a transit agency down in Portland, Portland Tri-Met, and some years ago they were faced with the question of whether they should renew the data contract they had for the center-line data that fed their transit maps and routing systems. And they made what was, for the time, a startling decision.



- Data vendor
- Moderate ongoing cost
- Low frequency of update
- Familiar, "low risk"

They saw they had two options, renew and deal with the rather long update cycles the vendor provided, or instead move to OpenStreetMap. That was not a simple option, because the quality of OpenStreetMap in their area at the time was not acceptable for either their mapping or routing needs.



T R I 🙆 M E T

But they saw that the incremental investment in bringing the quality up, for their area, would be less than the cost of renewing their data license, and that once they had expertise in OpenStreetMap editing they could add in updates far faster and cheaper than their old vendor.

So they hired a couple interns for the bulk updates, and trained their staff in OpenStreetMap editing, and dropped the vendor.

Open Street Map Plan

Open Street Map Plan

Upgrade existing data (2 interns)Maintain upgraded data (Staff training)

- Upgrade existing data (2 interns)
- Maintain upgraded data (Staff training)

Open Street Map Benefits

- Better internal sharing no license issues
- Better external sharing OSM distribution

In addition to getting better, cheaper and more up-to-date data, they also increased the quality of the local data, which in turn increased the quality of data available to all the other organizations in the region.



Even applications over which they had no control and even no relationships with, those apps and users I mentioned earlier, Strava, Nike, Snapchat and so on, all have better maps of Portland, because of the TriMet decision to join the OpenStreetMap community.

What's the best way to distribute public data? Where's the best place to manage public data?



Now, think about the data on your open data portal, and ask yourself, is it better there, or would it be even **more valuable** if you integrated it directly into OpenStreetMap. Where are your citizens more likely to derive value from it? As a shapefile on a web site, or as data in the global open map of the world?



I feel like I'm leaving a lot on the table here, that there's a great deal more to talk about and not enough time to fit it all in.



In the title to this talk,

I promised to "Modernize Your Geospatial Practice with the One Neat Trick"

and you're probably wondering what the heck that trick is.



The big picture of the changes I have described, in how software is built, in how large bodies of knowledge are built, is one in which hierarchical organizations — corporations, government, the military --



are overtaken and left behind by collaborative networks of individuals.



The trick, and it's a difficult trick to carry off, is to actually engage directly with these new communities, because that involves giving up the illusion of control, it involves abandoning hierarchical authority, and engaging as equals with people who might in other contexts be marginal.



You'll have to start by figuring out open communities, first just by using the tools and the data they make available, as a consumer.

But the next step, the transformative step, it to engage with them as community members.



Or, more correctly, to encourage, to incentivize, to reward, your staff for engaging with them as community members.

Open communities are networks of individuals, not corporations or governments

There are no MOUs and contracts, there is a social contract which people have to engage with.



If you can nurture, in your organization, the kinds of people who engage with these communities, who can communicate the values of the community back to you, and communicate your organizations goals back to the community, who can build bridges between the two,

you'll be well on your way to transforming your geospatial practice.



Because those people, those geonerds and geogeeks, will get more out of their jobs, as individuals, and bring more value back into your organization, as members of those communities.



It's a neat trick, and it's not an easy one, but it's one that will pay dividends for years to come, in technological flexibility, in staff engagement and career growth, and in the long run organizational relevance to the citizens we serve.



I hope you'll consider giving it a try. Thanks,