

Why QGIS should be part of everyone's GIS toolbox

 millermountain.com/geospatialblog/2017/04/20/qgis-arcgis-users

Istomsl

April 20, 2017

QGIS is an open source desk-top GIS program. It plays the same role as ArcMap in the ESRI ecosystem. QGIS even comes with QGIS Browser, which is similar in function to ArcCatalog. Most GIS analysts in the US learn ArcGIS in college and work for companies that have ArcGIS available. Many people are under the impression that if they have access to ArcGIS, there is no advantage to their company for them to learn QGIS.

I felt the same way for over a decade. Over the past several years I have come to believe that QGIS has many advantages over ArcGIS. This is true even for companies that already own ArcGIS licenses. There are some things QGIS does much better. Some things it does much cheaper. And admittedly, there are some things that ArcGIS does better. The important thing is to understand the differences and when to use QGIS and when to use ArcGIS.

Open Source is not “free-ware”

First a word about the term “open-source”. If you are as old as I am, you probably remember the days of “free-ware” and “share-ware”. These were business models where individuals wrote software and released it to the public for free. Sometimes it was out of a sense of altruism. Sometimes the intent was to provide a subset of features for free and then charge money for a “full” version. Other times the intent was to include advertisement for money, or attract users to other products. When I first heard that QGIS was open-source and free to use, I thought of it in terms of the “free-ware” model. But I was wrong.

What is open source?

Free-ware is usually designed by one person and generally released with an ulterior motive to earn money in some way. Open source is very different. Open source is better thought of in terms of “crowd-sourcing” rather than garage-tinkerer. For example, Wikipedia is an open-source encyclopedia. It's not perfect but it is very complete and functional. It would be difficult to compete with it using a for-profit model.

Some open source software projects are large enough to have a foundation with a committee overlooking the software. The committee may even have funding from donations and corporate sponsorship. Open source software relies on multiple programmers contributing features and fixing bugs. For free.

Why would anyone donate either money or time to an open-source software project? The answer, in my opinion, is that open-source development has proven to be a model that results in stable software that just simply works well. There is no marketing department to demand fancy bells and whistles. It's just a bunch of computer geeks working together to come up with something that works. Many people get frustrated with software that has a lot of features but struggles with the basics. Open source is the opposite of that and people are willing to contribute time and effort to create that software for themselves.

Who uses open source software?

You do. And so does everyone who uses the internet. Much of the web is driven by open source software technology. Linux is an open-source operating system that runs much of the web. The Apache web server is open source and handles over half of internet traffic. MySQL, PostgreSQL, and MongoDB are all open source databases that handle huge amounts of web data. JavaScript and PHP are open source programming languages that are used by the vast majority of web-sites.

In fact, everywhere you look, there are open source projects that are increasing in popularity and dominating their fields. I already mentioned Wikipedia. R is an open-source programming language that has become the dominant tool of statisticians and data-scientists. Python is an open-source programming language for more general programming that is also very commonly used in Desktop GIS software. The point is that open-source software is not a toy, like many free-ware projects were. It is often very complicated, very stable software that simply works well.

How is QGIS different than ArcGIS?

Although ArcGIS and QGIS both occupy the desktop GIS realm, there are some differences. In general, I think ArcGIS is a more polished product, at least externally. ESRI has the advantage of having a team of paid programmers and a top down management structure. This means that, most of the time, there is a clear and consistent methodology for addressing a problem. There are exceptions however. For example, ESRI's approach to web and mobile GIS over the past decade has been downright schizophrenic. ArcGIS also tends to have more features that I would describe as bells and whistles. These are features that are great if you need them, but are rarely needed in general GIS applications. The need to sell software and convince users to upgrade to the latest and greatest version requires a steady stream of new features.

Open-source software on the other hand, takes a bottom up approach. Anyone who wants to take a stab at fixing a bug or adding functionality can do so. QGIS allows plug-ins that individuals can contribute to add features. As a result, there can be several methods available for addressing a problem and those methods are often of varying quality. Therefore, a little more work is required by the user to find the solution that works best for

their needs. On the other hand, this approach also means that it is more likely that the user can find a solution that more closely fits their needs. Open source software is user-driven rather than market driven so the focus is on software that works well, rather than adding bells and whistles in order to sell product.

One of the biggest advantages of open-source software like QGIS is that it is free. Of course, that has to be tempered with the cost of learning new software. If you are a power user of ArcGIS, you may not want to invest the time into learning new software when you already have access to ArcGIS. Many companies, however, have a wide variety of GIS skill levels within their organization. Often there are a few power-users for whom GIS is their primary focus and/or interest. There is also often a much larger number of casual GIS users. For instance, project managers often want to use GIS to view data that their team produces and create the occasional map for a report. Lower level field personnel may need to use GIS to enter data they collected or review existing data prior to a field visit.

In my experience, it is often cost-prohibitive to provide an expensive license for every GIS user in a company and team members are rarely happy about not having GIS available when they need it. QGIS can fill in as a lightweight data viewer/editor for these casual and occasional users. Many of these casual users are not very comfortable with ArcGIS in the first place and thus, switching to a different software package would not present a major challenge to them.

There are also some things that QGIS flat out does better than ArcGIS in my opinion.

1. QGIS is available for Windows, Mac OS, and Linux. There is even a port of QGIS for android in the works.
2. I think that the editing in QGIS is much more straightforward and easy to use than ArcGIS past or present. I've been told that ArcGIS has more advanced topology tools available and I have no reason to doubt this is true, but I've never personally needed any editing functions that I couldn't find in QGIS.
3. QGIS has a much broader range of data formats that it can read and edit. This is especially true for formats that compete with ESRI. For example, SpatialLite, PostGIS, GPX, and GeoJSON support is included out of the box. In fact, QGIS is worth downloading just for the purpose of translating between data formats.
4. Some functionality that requires expensive extensions or advanced licensing levels in the ESRI world are available out of the box in QGIS. For example, map algebra, creating contour lines and hillshades from DEMs, simplifying polygons, etc.
5. Open source software places a greater emphasis on backward compatibility. Solutions that you develop are less likely to become obsolete the next time a new version comes out. In fact, unlike ArcGIS, you can have multiple versions of QGIS running on the same machine if you need to for backward compatibility. The days of frantically uninstalling a new version of Arc and re-installing the old when it doesn't run some custom code or someone else's plug-in are over.

6. Multi-user editing capability right out of the box. This is huge in my opinion. In fact I saved it for last only because I felt it warranted an entire section or two on its own.

What is “multi-user”?

Multi-user means that multiple people can access the same data. ArcGIS supports multi-user viewing of the same file, however it's support for multi-user editing is very disappointing in my opinion.

Before I go any further, I want to say that I think ESRI is a great company. ArcGIS is a great product. Jack Dangermond did a great service to the world in developing the first commercial GIS software. I do not have a problem with ESRI making commercial GIS software as a for-profit business. In my opinion, however, their refusal to implement affordable multi-user editing in ArcGIS is a huge mistake. It is driving many ArcGIS users towards open-source solutions. I was one of them and quite frankly having gone there, I see no reason to ever go back. Open-source software works great for 95% of what I need GIS to do for me and for the companies I work for.

Prior to the year 2000 most GIS data was stored in either ArcInfo coverages or shapefiles. With the advent of ArcView 8.0, ESRI introduced the concept of the geodatabase. A geodatabase extended existing database technology to store spatial data. At first ESRI geodatabases came in two flavors, personal and enterprise. Personal geodatabases were built on the Microsoft Access mdb file structure. Enterprise geodatabases were built on existing enterprise level multi-user databases such as SQLServer or Oracle. These are the databases that large corporations use to store their data and they can handle millions of users around the world via the internet.

Using Access files for single-user geodatabases was limiting due to a 2GB maximum file size, so ESRI developed its proprietary file geodatabase specification. File geodatabases store data in multiple files within a single directory. They are great for single user applications. They are efficient in storing data and optimized for data transfer with spatial indexing. Unfortunately in my experience, they do seem to have a tendency to get corrupted, probably due to the large number of individual files they are composed of. Enterprise level geodatabases are great for large companies that already have an existing database system and the ability to hire staff with the IT expertise to setup an enterprise level GIS

The problem, in my opinion, is that they left small and medium-sized companies high and dry. Many small to medium sized companies reach a point where they need multi-user editing capability on occasion but the cost for moving to enterprise level geodatabases in the ESRI ecosystem is 10's of thousands of dollars. In my experience, that is a very hard sell for a GIS analyst to make to the owner of a small to medium sized consulting firm. It is also a hard sell to make to clients.

Even worse, I found it very difficult to get a straight answer from ESRI on what the costs would be. The \$5,000 for a basic ArcServer license doesn't sound too bad but that only covers 3 editors. Anyone who is editing also needs an ArcEditor or ArcGIS Standard (or whatever they call it these days) at 7k each. That doesn't include the cost for installation, training, migrating data, etc. And if you need more than 3 editors you need to purchase SQLServer or Oracle on top of it all and get your IT staff involved. If you're a GIS analyst without extensive IT background, it's enough to make your head spin.

What could have been....

It didn't have to be that way. There is no reason that ESRI couldn't have developed a simple system with limited multi-user capabilities. Microsoft Access, for instance, allows record-level locking. This means that only an individual record (or feature in GIS terms) is locked when someone edits it and the lock is released as soon as the person is done editing. Access is a far cry from an enterprise level database but it can handle a reasonable amount of multi-user editing through this system and ESRI already had a geodatabase built on the Microsoft Access file structure.

SQLite is a file based database system that sits in the gap between a single-user database such as Microsoft Access and a full-blown enterprise database. It does lock the database during write operations but the lock is released immediately in milliseconds and so is barely noticeable for small projects. It's even open source so ESRI could easily have used it as the basis for its "single-user" geodatabase and would have had pretty good multi-user capability for free. Again, not full-blown enterprise level capability but more than enough for any of the applications that I needed in my 15 years working for three different environmental consulting firms. In fact, there is a very useful open source geodatabase built on SQLite called Spatialite that QGIS and other open-source GIS software can use.

Even if ESRI decided SQLite was not the right tool and insisted on developing their proprietary file geodatabase, they could have made even a slight effort to make it easier for multi-user editing needs. Instead, they seem to have gone out of their way to make it impossible to use file geodatabases in even the simplest multi-user application. For instance, they could apply a file-lock ONLY when a user goes into editing mode and release the lock as soon as the user exits editing mode. This is in fact what multiple levels of ESRI phone support people told me they did. It turned out not to be true. After hours of waiting on hold and waiting weeks for callbacks, I finally talked to someone with enough understanding of the problem and honesty to admit to me that that is not how it works.

What happens is that the first person to open a feature dataset has exclusive access to editing that feature dataset until they exit ArcGIS. This happens whether they are editing anything or not, which is ridiculous in my opinion. A project manager who comes in early

and opens a map locks up all the data in that map so that nobody can edit it until they exit. Worse, there is no way to find out who, among 10 or 20 potential users, has the lock so you can't even ask them to exit.

I'm not privy to ESRI's inner workings. I don't know why they chose to do it this way, and worse, why they continue to do it this way. Certainly, there is no technological reason for it. Solutions exist and ESRI has smart developers working for them who could solve the problem if directed to. The only explanation I can come up with is that it was a marketing decision to push people towards spending 10's of thousands of dollars on enterprise GIS. If there is another explanation, I'd love to hear someone explain it to me. I can only say that for me, as a GIS analyst for environmental consulting companies with 20-30 people, that decision failed. If it were up to me I would have stayed with ESRI, simply for simplicities sake. I couldn't do that, because my employers would never have spent the money on ESRI enterprise geodatabase software. ESRI's refusal to find a reasonable solution for moderate levels of multi-user editing pushed me into looking at open-source solutions. I found PostGIS and QGIS which fulfilled all my multi-user needs for free and my employer was very happy.

What is PostGIS?

PostGIS is a spatial extension to the PostgreSQL database. In the ESRI ecosystem it serves the same function as ArcSDE. Since PostGIS and PostgreSQL are open-source, the software is free. There are, of course, costs associated with installation, training, migrating your data, etc but those costs will exist even if you stay with ESRI. The primary purpose of PostGIS is storing spatial data but there are also over a 1000 spatial functions included that can perform a lot of spatial analysis, conversion, and data management through SQL. For instance there is a buffer function that allows you to create buffers on the fly, rather than actually storing the buffered features in a table.

Because it is built on the enterprise database PostgreSQL, PostGIS is able to handle thousands of users simultaneously and even serve data over the internet. I have built several web mapping and mobile data collection applications that use PostGIS as the backend to store, modify, and retrieve spatial data.

QGIS works with PostGIS natively. In fact, QGIS was developed partly to serve as a GUI interface to data stored in PostGIS. You can have users anywhere in the world editing data through a web or mobile application that uses a PostGIS database and their edits will be available in QGIS in real-time. Likewise, you can edit PostGIS data using QGIS and those edits will be available to users anywhere in the world with an internet connection instantaneously. And you can do all of this without paying for anything other than the cost to develop the applications.

Why do I need multi-user editing?

The short answer, is that I don't know why YOU need multi-user editing but I believe that most small consulting firms will eventually benefit from it if they want to remain competitive.

Here are three examples.

1. A couple years ago the company I worked for got a contract to create NWI wetlands data for the Navajo Indian Reservation. This involved looking at color infrared photography of the entire Navajo Reservation at 1:6000 scale and digitizing wetland polygons that were identified through aerial photo interpretation. At that scale it required 30,000 separate views and I developed a tool to track everything to ensure that each section was looked at and went through a QA/QC process. Everything worked great as I was developing it but as soon as we had more than one digitizer we ran into problems. ArcGIS just couldn't handle multiple people editing the same feature class. I solved the problem in this case by splitting the area up into separate feature classes, one for each of our digitizers and giving each digitizer a custom map that only linked to the feature class that contained the sections that they had been assigned to. This created a lot of extra busy work for me as I had to split the area up, assign each area to a digitizer, and merge them all back together again. Then I had to repeat this for our QA/QC process. The additional workload wasn't nearly enough to justify the \$70,000 it would have cost us to implement an ESRI style enterprise database. If I had been aware of my options at the time, however, it would certainly have justified the effort to implement the whole system in a PostGIS database with a QGIS front-end.
2. Another company I worked for had a couple large oil and gas customers in an active oil and gas field. Our job was to identify and monitor environmental constraints that might affect their operations. Our field crews would go out every morning and conduct surveys and then return every afternoon and add their findings to our database. It was a constant battle and source of frustration for everyone involved to come back from a long day in the field and have to wait in line to enter their data in the afternoons. It was also limiting to the project managers who didn't have access to their data during that time period. I addressed the problem primarily by people management and making sure that everyone in the office exited ArcGIS before the field crew came back in the afternoon and then having them enter their data one at a time. It was definitely not ideal but still better than spending \$50,000 on an enterprise level GIS system.

3. Another company I worked for did a lot of mosquito control work. I was tasked with developing a system for our field crew to record information about their inspections and treatments in the field. This time around I didn't even consider an ESRI based solution. There was no way we would be able to afford \$500/per tech/per year for an ArcGIS online license so they could use Collector. In fact, we had a lot of needs that Collector couldn't address, and quite frankly I had poor results in the past when using Collector in off-line mode and I was hesitant to deploy it. When we added up the cost of ArcGIS Online subscriptions for 50 techs, year after year, we quickly realized that would be far more cost efficient to develop our own system that did exactly what we wanted, using a web interface to PostGIS data and using QGIS as the desktop client for office staff. This system had no trouble handling 50 field techs entering data all day and our only cost, other than development, was \$15/month for web hosting.

Why would I even keep ArcGIS?

You may be wondering, If QGIS is so great, why would you even use ArcGIS? If you're like most GIS analysts, you probably already own a copy of ArcGIS or have access to it at work. You probably learned ArcGIS at school and you're probably intimately familiar with its nuances and quirks.

I've been impressed with how easy it was to do most things in QGIS. Making the switch wasn't as painful as I expected but I won't lie to you. There is a learning curve. It will take a bit of time to become as efficient with QGIS as you are with ArcGIS. And there will probably never be bells and whistles such as ArcScene or ArcGIS Online (Although QGIS 3.0 has a limited [3D viewer](#) and you can create some pretty decent client-side web maps with the [QGIS2Web](#) plugin). If you use these parts of ArcGIS, you'll probably want to continue using them. I still find myself going back to ArcGIS for some simple things when I'm in a hurry, although less and less every month. And I have had QGIS choke on some large geoprocessing operations that ArcGIS handled with ease. (I've also had QGIS handle some geoprocessing operations that Arc choked on, so the answer as usual is.... It depends).

The bottom line for me, is I would never want to be without both, even though I find myself using QGIS more and more. I view them as a part of a quiver of tools, the trick is learning which tool is the best to use for any given situation.

What's the optimal architecture?

There probably isn't one answer that will be optimal for all companies. If I were starting a new company, I would base it around QGIS as the frontend and PostGIS as the backend. The advantages of being able to develop web interfaces to data stored in PostGIS free of cost and to have multi-user editing are so great, that I would never want to be without them

in today's hyper-competitive consulting market. I would also have at least one ArcGIS license and probably 3D analyst as well for myself and maybe a few others for power users, but everyone would be accessing our data primarily through QGIS.

Most of us won't have the luxury of starting from scratch, nor will we have the luxury of having large amounts of time to dedicate to making a transition. I would suggest giving everyone access to QGIS and maybe having some internal training. There's nothing like some live demonstrations to make people think "Hey, that doesn't look so hard". I would suggest starting new smaller projects with QGIS to get familiar with it. If you've been struggling with trying to get around ESRI's restrictions on multi-user editing on larger projects I would start planning a transition strategy for those projects as well.

If you get new projects that would require multi-user or mobile access or other features that would require spending more money on licensing (i.e. for spatial analyst, etc) I would definitely evaluate to see if QGIS/PostGIS could fill those roles without the need to spend extra money. Its usually easy to sell management on giving you time to learn new software if they see it as saving them money on the long term. Especially saving money on software which is not as much fun as shiny new trucks (at least in the eyes of managers).

It gets a little tricky, because ArcGIS can display data stored in PostGIS but can't edit it. You can however use them as inputs to geoprocessing operations as long as those operations don't modify the original data. QGIS on the other hand can easily display and edit data in a shapefile, and can display data in a file geodatabase but cannot edit data in a file geodatabase. There are plug-ins available for editing file geodatabase data but my luck with them has been mixed. You can set-up a hybrid system, where one set of data is accessed by both ArcGIS and QGIS but you should probably only attempt to use one piece of software for editing and the other for viewing. I would suggest storing the data in PostGIS and editing in QGIS for a number of reasons.

1. You can have as many editors as you want or need for free.
2. I think editing in QGIS is more straightforward than ArcGIS.
3. The things that ArcGIS does really well, like production quality maps, data driven pages, ArcScene, Model Builder and some geoprocessing operations don't require editing.
4. If you ever need to develop a web interface to the data or a mobile data collection system that directly accesses it, you can do this far easier and cheaper with data stored in PostGIS.

The main exception to this rule would be if you needed advanced topological editing tools that weren't available in QGIS.

Where to go from here (Sales pitch ahead....)

In summary, I would argue that the benefits of knowing how to use QGIS far outweigh the cost of learning a new software package. This is especially true if you see the need for multi-user editing and/or web and mobile access in your future.

If you would like to learn more on your own, I have a course available on Udemy entitled QGIS 3.0 for GIS Professionals that will provide you with the tools you need to work with QGIS.

I expect to have a course available shortly (approximately Feb 20, 2018) on using PostGIS with QGIS that will show you how to implement a full multi-user enterprise GIS with PostGIS as the backend and QGIS as the front-end.

And if you are interested in learning to develop web based GIS applications with open source software I have an entire series of courses available for that.