**Testable Requirements Statements**

Example of a bad requirements statement:

* ArcGISPro won’t work
* We want PostGRE GIS
* Desktop software is not adequate
* That stuff is tinkertoys

You need three things for performance.

1. Server Operating System
2. Hardware
3. Stack

Items we still need to turn into good statements:

* OGC file format and metadata
* mainstream linux distribution (Debian, Red Hat, etc.)
* PostgreSQL + PostGIS
* GRASS GIS
* R
* MS SQL server for keeping current applications running
* Mapserver
* GDAL
* RStudio Server
* Apache WWW server

Here are some things that I think would help:

1. Workflow requirements – Do you have the need for workflow features like approvals, routing work, alerting, conditional branching of work, or returning work when not approved? I know this would be future needs.

Yes, this will help Jason with requests, getting approvals, sending to Skye or others for help, letting customers know if their product is approved by peer review, tagging for release to public, etc.

1. PB tabular and spatial – I think it would help a lot if you could share a back of the napkin level explanation of the size estimate of DSP.  For example, soil data is 50GB and for each predictive model we will generate 100 data elements and we expect to do 1000 predictions.

You two will have to come up with these (I cant even guess!)

1. The main thing I’d like to see is the algorithm for determining how the input factors relatively contribute to the outcome.

I am not sure I understand what he is asking.  Is this statistical analysis visualization? Relative influence statistics? Regressions?

Possible new requirements from today’s talk: These look awesome – you guys will have fun with these!

* Ability to experiment with DSP data on multiple machine learning/predictive engines/fuzzy logic to support exploring developing algorithms and automation to predict outcomes under conservation practices
* Because DSP requires multiple artificial intelligence, analytic, and statistical tools, DSP requires a user interface and automation to reduce manual work interacting with multiple disparate tools (platform to tie everything together)
* DSP requires a dedicated tabular and spatial development database that can be administered by end users with full system administration privileges as a working area to support development of DSP

PostgreSQL + PostGIS

Robustness – Any enterprise level database software, including PostgreSQL, is going to be orders of magnitude more robust (safe from crashes, data corruption, etc) than a file-based data storage. Commercial database server companies compete on the basis of robustness, security, and speed. If their databases fail it can be national news and billions of dollars are at stake so they have the best and the brightest software engineers, they can hire making sure that any possibility of crashing or security glitches are minimized. And open-source database server projects are not much different. In fact, most of the web is based on open source database servers because of their economic feasibility.

Speed – Many GIS operations can be performed in PostGIS itself, and since PostGIS is running on a server and servers are generally much more powerful than a desktop computer you may see significant performance gains and may be able to purchase less expensive computers for GIS users in your organization.

We need high-performance, multi-threaded software and libraries that can scale to very large vector, raster, and tabular data sources. This software should by fully compliant with OGC file format and metadata.

SQL Server sitting adjacent to PostgreSQL has demonstrably better spatial handling, indexing, manipulation and serialization capacity. Its ability to work with R vector data is more efficient to what we can do now. SQL server and Postgre don’t interfere with each other.

The DSP hub is similar in many ways to the NASIS modernization. The DSP hub will be   transactional as well and even more complex as far as aggregating, modeling and the querying capabilities.  Data from the DSP hub would be imported from NASIS but put back into NASIS.  It’s transactional in a way that it will be constantly being manipulated. Data would be imported and entered in the DSP hub.  It’s no different than the NASIS system but far more data hungry and processing power. If the NASIS modernization isn’t in the data lake, why would the DSP hub be in it as well.

PostGIS + PostGRE SQL

With a spatial database working with large datasets becomes possible. Not only easier, but sometimes it’s almost impossible to work on larger datasets without a database. Have you ever tried to open 2 gb csv file? Or tried to do some geoprocessing for a 800 mb GeoJSON? Did you even know that Shapefiles (ESRI product) have a size limit? Of course, you can tackle some of these issues by using Geopackage or some other file formats, but in general PostGIS is the optimal tool for handling big (geospatial) data than using ESRI ArcGIS. A good example is ESRI portal and CART. There is lag and compatibility issues with SQL server when using just one layer.

As a rule of thumb, the more data a query has to fetch and more operations the database has to do (ordering, grouping etc), it becomes slower and thus less efficient. An efficient SQL query only fetches the rows and columns it really needs. SQL can work like a logical puzzle, where you really have to think thoroughly what you want to achieve. Under the ESRI framework this isn’t really achievable.

Linix

Having it on a linix distribution we cut out a tremendous amount of overhead that has occurred by the operating system. You don’t need a graphical interface. This is machine is made for crunching data and request that come over the network.

Example of a good testable requirements statement:

Decent Testable Statements for Dylan and Jason to review:

* We need to be able to process multiple soil geospatial layers at the same time without \_\_\_\_ loss in performance.
* We need to have a separate/unique rules engine that is for the DSP Hub workbench
* We need high-performance (need a specific processing ###),
* Software that can efficiently utilize as many CPU cores as needed (48 or 64 core)
* Max RAM – 512 GB for typical analysis
* RAID (Redundant Array of Inexpensive Disks) – spatial data analysis and operations bottle neck is the disk
* DSP Data Hub is expected to grow to up to 1000 PBs
* DSP Data Hub will contain more than 100 quintillion records of data
* DSP requires a computational engine that can process 100 trillions of transactions per hour
* Due to the large, complex transactional needs, DSP computational engine is required to run server side (analytic tools on client-side would not meet transactional volume nor handle the size of the data)

Ability to handle analysis of files such as:

* Vector files – typical size 200 Gigabytes each, which typically would have files analyzed at one time
* Raster files – typical size 2 Terrabytes with potentially 150 analyzed at one time
* Tabular data files – typical size up to a Pedabyte, which would need to be analyzed with other raster and vector files

Conclusions

If your way of working is currently inefficient, just changing your tools won’t make your outcome any better or the process less painful. You need to change the way you think about data management. There are numerous ways to use databases inefficiently. Trust me, I have seen them and even tried a few.

Also changing things just for the sake of change, doesn’t make sense. If your daily work is just plotting a few dots on a map every now and then, you can very do that with Shapefiles and csv files also in the future. Might even be more efficient that way.

BUT.......

If you do want to do some serious spatial analytics, automate your processes or in any way move your way of working with spatial data to the next level, using PostGIS and especially spatial PostGRE SQL is the way to go.

References:

1. http://blog.cleverelephant.ca/writings

2. https://geo-bigdata.github.io/2015/papers/S08206.pdf

3. https://arxiv.org/pdf/1907.12182.pdf