04\_23 MVP Questions\_consolidated.docx

Comments are preceed by [name] rather than showing colors. By “today” I am referring to our SQLite meeting 4/23/2021.

1. Potential installation of SpatiaLite DLL(s), Python and Python libraries may require admin or USDA/CEC assstance.

[Kyle] If installation of DLL(s) are required than we will need to consider altering our MVP. Give that spatial will be dealt with in a later phase, we can deal with this down the road but leave the door open to modify the requirements. 6/28/2021 - If we require Jupyter Notebook to execute python scripts, then we run the risk of customers requiring advanced/elevated/admin privileges or requiring IT assistance. It seems to me that this is inevitable and acceptable because we need an application to execute the scripts, which Phil pointed out from the very beginning. The MVP should be updated to reflect business’s acceptance of this potential requirement.

[Steve] In my experimentation, I have been able to use the standard version of Python (3.7.9) and standard set of libraries that are part of the ArcGIS Pro 2.7 distribution to do all of my work. I am able to create databases and create all attribute tables and constraints using the included sqlite3 library. For the spatial import I have been using arcpy, which works fine with ESRI software, but I plan to replace the spatial import with one of the other non-ESRI Python libraries. Phil, this is basically where I’m at now. I’ve been focused on using Spatialite DLLs and Python OGR libraries to try and replace the arcpy library and that has been a painful experience (DLL Hell). I suspect that one of the non-standard (for ArcGIS Pro installation) Python-Spatialite libraries may work better but I’ve been trying to avoid altering my ArcGIS Pro environment. I wouldn’t mind seeing some additional research on this subject to identify what specific DLLs or Python libraries would be required as part of a software/database distribution.

[Phil] ESRI already includes DLLs. Their inclusion (for example SpatiaLite and required Python library files) very much depends upon the user’s environment.

2. Policy and mechanisms for managing hybrid content (e.g. data from multiple sources and/or SSA plus WSS ad-hoc AOIs) is not yet explored or defined.

[Kyle] The MVP may already address this point. The MVP states “Database can store multiple soil survey areas”. The source of the SSA shouldn’t matter but the database should be able to store multiple records in the legend table and other tables that store SSA information. This should be true regardless of how the SSURGO data is generated and regardless of how many SSAs are included within a single set of SSURGO text files.

[Steve] Totally agree. We might want to test the ability to import multiple WSS-AOI’s from a single soil survey area into a single Template database. I’m not sure if the current SSURGO constraints can handle that or if we should even allow that.

[Adolfo] I’m still a bit puzzled by this “IF” the different input sources have different database structures. Are we assuming that any source input will have the same structure as a SSURGO download (.shp files and .txt tabular files)? If so, then disregard this comment. If not, how many different input formats/structures should be accounted for. The 2 different sources that come to mind are the WSS downloads and an export from NASIS to QA/deliver preliminary “unofficial” SSURGO datasets. Is the NASIS export structured different than WSS?

[Phil] There are two different facets to “hybrid”: data source and SSURGO “AOI” type. If we do not require numeric keys then data source (Nasis, Staging, WSS) is likely to trigger a key collision. WSS offers two kinds of AOIs and therefore two SSURGO packages: an SSA package and a user’s Ad-Hoc package. I don’t see a problem however an Ad-Hoc AOI may include partial data from multiple SSAs, collision between hybrid WSS SSURGO data sets may present some interesting situations. 6/28/2021 - I can’t foresee any issues if we need to blend SSURGO from Staging/NASIS/WSS. The legend table will use the same primary key (lkey) value for all three SSURGO tables. That is, in all three databases, the lkey is equal to the NASIS database legend.liid (legend table record ID). This is what we want in the data structure so soil survey areas can’t be duplicated in the SQLite database, regardless of source of SSURGO. For example, if I create a state wide database for Oregon from WSS SSURGO but then decide to incorporate data from NASIS, I should receive a message if I try to import a Oregon Soil Survey Area (lkey) that already exists in the database. Furthermore, looking downstream, or I should say, down child table, NASIS and Staging uses concatenated primary keys for component table and all child tables. This means there is no chance of primary key form NASIS/Staging conflicting with the keys assigned to the SSURGO data derived from WSS. That would only be a problem if we were to change the NASIS/Staging keys from concatenated to integer.

[Steve] I’m not sure that we should be supporting importing multiple WSS-AOI downloads into a single database. There have never been any support calls for this type of thing, although we have had several requests for the ability to merge multiple state gSSURGO databases which sometimes overlap. I suspect it could work if the import process handled all data one-record-at-a-time using the sqlite3 library and skipped past any that generated unique constraint errors. There would be a performance cost. I currently do this for the sdv\* tables in gSSURGO.

Currently the SSURGO Template database ignores any of the thematic map information that is included in a WSS-AOI download. That functionality is something we should add to the list for future development.

3. Use of metadata text files distributed with WSS SSURGO downloads is not considered.

[Kyle] Let me ask this, do we even need to consider these other than ensuring the metadata text files included in the SSURGO download are up to date? Here is another thought, do we even need to retain these files? Other than Steve using them for gSSURGO schema, what purpose do they serve and are they used by customers? If there are select group of people that do need these files (e.g. Steve for gSSURGO), could we simply redirect them to the SQLite database as the source of the metadata and remove these metadata SSURGO text files from the system. That results in one less piece in the puzzle.

[Steve] When we originally began creating and distributing soils data in the gSSURGO file geodatabase format, I incorporated a static copy of all the ‘mdstat’ tables obtained from the MS Access database (US\_2003 version). After I few years we switched to using a template gSSURGO database with empty ‘mdstat’ tables and imported the tabular data from the textfiles. The reason for this switch was because I discovered that some of the domain values in the mdtatdomdet and mdstatdommas tables were not being kept up to date in the original Access Template database. This resulted in an apparent discrepancy in the some of the maps and reports generated by Soil Data Viewer vs Web Soil Survey. I think maybe it was an issue with the AASHTO domain, but I could be wrong.

A few additional comments: 1. All of the System-SSURGO-Metadata reports in the current SSURGO Template database use the mdstat\* tables. 2. If it weren’t for the mdstat\* textfiles, I would not have had any way to know that some of domains and column definitions were out of date in the SSURGO database. I think it would be a mistake to assume that we will always keep the template database up-to-date and version numbers up-to-date in the future, even if we move the process into NREPO. We don’t have a Dorn available to keep track of this stuff any more. 3. The mdstat\* tables allow us to maintain a database-independent record of the database structure. Currently each database we are working with (SQL Server, MS Access, ESRI file geodatabase, SQLite) requires a different method to generate schema reports) and some of those methods are rather involved.

[Adolfo] The ‘Generate SSO SSURGO Datasets’ tool in the SSURGO\_QA\_Toolbox uses the metadata files to import tabular data into a FGDB. This tool creates MLRA extent SSURGO FGDB and is used by every Regional GIS Specialist. Aside from Steve’s tool and my tool, I don’t know who else uses the \*md metadata files. I occasionally use them as a reference, but I can use the SSURGO metadata pdf files as well. Wouldn’t the SQLite database use these files in similar capacity as SDV used the md files within the access template down the road? Kyle, if the \*md files already existed within the SQLite database as part of the distributable template then I do think we can remove them from the SSURGO download. Regardless of who uses them or how they are used they should be up to date.

[Phil] I’d rather that we considered the “version.txt” as a proxy for the metadata version, rather than reading the md\*.txt files and performing a comparison of the md\*.txt data. I’d prefer to eliminate these from the WSS download for SQLite SSURGO content.

4. Definition and delivery of an “empty template” is not finalized. The stories define a path to a manually-created empty template. We do need to refine this.

[Kyle] My thinking was that we deliver the same sort of product we deliver for the mdb. That is a database that has the metadata tables and payload tables, along with relationships, indexes, and constraints. The idea was the user downloads the SQLite template and runs some script(s) in some environment to load the SSURGO data into the payload tables. I think this needs further discussion because as I understand the process you proposed, this would not be the case. Instead, the database would be initially constructed or rebuilt by the python script that executes the SQL statements. That may be okay but the “empty template” was us saying we just wanted to do the a similar thing that we did the mdb. 6/28/2021 - I agree with Phil’s list and the need to clarify empty template requirements. The MVP should be updated to reflect the list.

[Steve] I would agree, at least for the MVP and the first version of the new Template. The look and feel of the Template should remain the same as much as possible.

[Phil] My use of the word “empty” is ambiguous. I suggest the following “empty template” containing the following:

1. All md\* tables.
2. All “payload” tables with no data (future: spatial data tables with “fid” or similar).
3. Additional spatial-support tables as suggested by the GeoPackage schema, including spartial reference information (like a shapefile’s “.prj” file) and a table of spatial columns.
4. Scripts (both SQL and Python, R, etc.) as needed to simulate desired MDB behaviors.
5. Other administrative tables as required.

5. Management of version mismatch (version.txt vs SQLite instance) is undefined.

[Kyle] I would like Steve to comment on this one.

[Steve] I’ve done some preliminary investigation and I sort of understand the concept of the how the database, tabular data and spatial data are versioned. This is a very important piece of the puzzle and deserves a lot more consideration than it has had so far. We haven’t had any changes in SSURGO versions for several years and so we’ve been spoiled. I do remember that the last changes in SSURGO tabular version and database versions and ArMap Addin version did cause a fair amount of support traffic, and those were pretty minor changes. I shudder to think what a major change could trigger if not handled properly and I suspect that even we handle it properly there will still be a lot of support issues. I think we should look at the versioning design and validation very carefully. I could see the potential for expanding upon the versioning design so that we can accommodate version numbers for template database type (SQLite, Postgres, etc), geometry type (ST\_Geometry, WKT, etc), raster type (GeoTIFF, Geopackage Tiled raster, etc). Bring on the Tylenol.

[Phil] My primary concern is with a mismatch in the md\* tables, where “version.txt” serves as a proxy for the version. Attemption to fuse data from a SSURGO version that does not mach the databases MD\* potentially introduces more “interesting” problems.

6. Ability to work with “non-ESRI open source programs such as R” is ill-defined. “R” is a language. What other languages beside Python and R are required? Is R in the MVP?

[Kyle] We should have said ability to work in other non-ESRI programs. I think ideally the new database could be used in R Studio but I know very little about R and I would defer to Dylan to better define R/R studio requirements. I think this language needs to be modified in the MVP.

[Steve] I agree that this area needs a lot more work and we need to write some simple stories that would describe our expectations for an MVP in the QGIS and R environments.

[Adolfo] Agreed, embedding flexibility such that non-ESRI software can interact with the new database format and its content should be the end goal. Nowadays users can download 3rd party libraries/modules for their choice of programming language to interact with an SQLite database. I know python and R can communicate with an SQLite database. The interoperability between a specific language and this database format would depend on what the user is trying to accomplish and makes it difficult to anticipate.

[Phil] In principle any sufficiently-broad “scripting” language, pared with a suitably-competent “application”, is conceivable. Permit me to reiterate: An MDB file is just a box of data. If you invoke it on a Windows system with Access installed the pair (application plus data) provides execution function plus data store. One could put an MDB file on an old PC running MS-DOS and no execution would occur. I will revisit this during our presentation. We will need to enumerate our expectations for both language and user-environment application.

7. Interoperability of a populated database is undefined. For example if data are loaded when using Arc GIS Pro, should the same database file be useable in QGIS?

[Kyle] I’m not sure how to answer this because I hadn’t envisioned using ArcGIS Pro to load data. The idea was to load the data outside of ESRI.

[Steve] Same as the answer to number 6. I agree that this area needs a lot more work and we need to write some simple stories that would describe our expectations for an MVP in the QGIS and R environments.

[Adolfo] The adopted database format should come with embedded scripts, shells, or an independent application to allow a user to import 1 or more SSURGO SSAs independent of a software. Once the data is imported we would need to decide whether custom tools are needed to interact with the imported data and a specific software to aggregate data and render thematic maps. Ideally, SQL views representing top 100 interps/ratings (i.e. example) would be included inside the SQLite database that can easily be called or executed. Any software that can communicate with the SQLite database should be able to read the results.

[Phil] This will be a rich area for exploiration. If time allows today I can show a GeoPackage loaded outside of any GIS system (I used the OGR utility “ogr2ogr” to load all spatial data from MT605).

8. Must the SQLite+SpatiaLite or GeoPackage be editable in ArcGIS Pro or QGIS?

[Kyle] I think it should be it’s not a requirement. Editing is rare occurrence and is greatly discouraged and should only be left to someone who is comfortable with the database.

[Steve] Agreed. I think at this time we should look at this new version of the SSURGO database as primarily a format for distributing the official SSURGO data and not as a platform for editing SSURGO data.

[Adolfo] Editing of SSURGO data should always be discouraged. However, one should be able to create new tables, queries…etc within the database.

[Phil] My concern is two-fold. If we want to allow export from SQLite then we need to define all target destinations. Does a target expect some fields to contain text not exceeding X characters in length? SQLite by default is weakly-typed, so an “integer” field could contain a text string. This coulkd be a problem for the recipient. The second issue is one of database definition: do we want to enforce data type checking and extend referential integrity.

9. Should spatial data (shapefiles) be loadable into the SQLite+SpatiaLite or GeoPackagewithout needing to resort to a GIS?

[Kyle] I think yes but I would defer to Steve, Adolfo, and Chad.

[Steve] I think for the MVP I would be OK with using ArcGIS Pro and perhaps QGIS to import SSURGO shapefiles into the new Template database. Ideally though we would want to develop a ‘GIS-software-independent’ method.

[Adolfo] Yes! The adopted database format should come with embedded scripts, shells, or an independent application to allow a user to import 1 or more SSURGO SSAs independent of a software.

[Phil] Demonstrated by OGR “ogr2ogr” as a first cut. I have not explored Python libraries yet for feasibility.

[Steve] 10. Business needs to write some stories on what we want the SSURGO Import process to look like. The original tabular import process had fairly limited functionality and was not what I would want, even for the MVP.

[Phil] The import process may vary with an application’s capabilities. I hope to demonstrate some of this today.