1. Create wrapper libraries
   1. Core features:
      1. XML Documentation
         1. Intellisense
         2. Help file
      2. Basic helpful enums & data types (help with documentation and parameter consolidation & reuse)
      3. Interfaces, to allow easier unit testing
      4. Internalizes 'numberOf' parameters where appropriate to reduce parameters.
      5. Simplified initialization with application
      6. Simplified plugin creation
         1. Template abstract plugin file.
      7. Throw exceptions rather than return error codes.
         1. If possible, exceptions have descriptions of the problem.
      8. Basic error avoidance/handling
         1. Catch API return code of 1, throw as exception.
         2. Check matching array lengths, throw as exception.
         3. Bracket limits to parameters where specified.
         4. Basic file ones in setting up API
   2. ~~SAP2000 (v19 only for now)~~
   3. ~~Add CSiBridge (v19 only for now)~~
   4. ~~Add ETABS (v2016 only for now)~~
   5. Connect into program:
      1. PluginCallback
   6. Later improvements:
      1. Add <para /> tags to all line breaks in all files?
   7. Later error checking:
      1. Comment out current extra thrown exceptions for integration testing. Re-implement through testing.
      2. Check behavior of "results in an error" cases in unit testing where it can be done within the method. Implement new thrown exceptions through this process. Note error checks that are multi-method to be handled in OOP wrapper.
2. Create OOP library (mostly inherited from wrapper)
   1. Core features
      1. Parameter objects
      2. Better consolidation into objects w/ state
      3. Better error handling?
      4. Add new functions as needed for simplified actions that require multiple function calls to API.
      5. Inheritance for generating design codes objects, etc.
   2. SAP2000
   3. CSiBridge
   4. ETABS
3. Create tests for wrapper library
4. Create tests for OOP library (mostly inherited from wrapper)