From Rich on 3/9/17:

Heilmeier's Catechism[[edit](https://en.wikipedia.org/w/index.php?title=George_H._Heilmeier&action=edit&section=2)]

A set of questions credited to Heilmeier that anyone proposing a research project or product development effort should be able to answer.[[5]](https://en.wikipedia.org/wiki/George_H._Heilmeier#cite_note-5)

* What are you trying to do? Articulate your objectives using absolutely no jargon.
* How is it done today, and what are the limits of current practice?
* What's new in your approach and why do you think it will be successful?
* Who cares? If you're successful, what difference will it make? What are the risks and the payoffs?
* How much will it cost? How long will it take? What are the midterm and final "exams" to check for success?

For the following, keep in mind there are new settings Model\_Mode can be set to for the integrated software:

* “MODSIM\_GSFLOW”
* “MODSIM\_PRMS”
* “MODSIM\_MODFLOW”

GSFLOW, PRMS, MODFLOW:

* Pass in model mode and enable the iteration time-step loop without MODSIM
* Compile PRMS as DLL (Steve, Eric, Enrique)
* Clean-up Fortran, specifically GSFLOW and various PRMS modules (Steve)
* Get PRMS-only Sagehen model running (Steve & Eric)
  + PRMS: Get lakes with multiple HRU’s running
* Add diversions to soil zone:
  + modsim2prms()
  + prms2modsim()
  + prms2mf()
  + sfr2prms()
  + wel2prms()
* Move budget/output routines inside GSFLOW\_PRMS() to proceed the RUN() for the next time step.
* 1st iteration of the 1st time step, all shared vectors are initialized to zero (except for the network mapping arrays)
* With Fortran MMF:
  + check that the RunTimePlots GUI for PRMS,
  + check to make sure the parameter tool works,
  + verify that RESTART() functionality still works
  + Long term goals for Fortran MMF:
    - Flexible dimensions
    - Multiple parameter files
    - NetCDF output support
  + Long term goal of allowing different time step lengths among the integrated codes.
* Add IUPSEG and IOUTSEG-like parameters to PRMS5
* Water-use option files need to be constructed so PRMS-only has them available. Use the PRMS format for this
* Does MODSIM still need to pass KPER and KSTP into the RUN() routines of GSFLOW (it looks like the various MODFLOW packages are passed this argument)? If not, consider removing from MODSIM communications with GSFLOW, just let the GSFLOW communications handle this.

MODSIM:

* Pass MODSIM diversions and releases to MODFLOW
* Pass MODFLOW accretion/depletion to MODSIM
* Pass in an Advance File Reader (AFR) flag
* Pass in “mapping” variables which are arrays of integers
* Pass back from GSFLOW the simulation\_start\_time and overwrite in MODSIM (this is different than data\_start\_time). MODSIM will need to ensure simulation\_start\_time and data\_start\_time make sense.
* Chunk-up to GSFLOW\_PRMS() into an INIT(), RUN(), & CLEAN() accommodate calls from MODSIM

Misc:

* Documentation

**Accomplishments at 1st code sprint (11/2/16-11/4/2016, Portland, OR)**

* Review Deschutes GSFLOW model
  + Provided technical support
  + NHM parameters transferred to his model (GIS overlay)
* Shared GSFLOW-MODSIM developments with:
  + OR WSC center director
  + State representative
* Jen and Bob presented the Deschutes MODSIM model (via webex):
* Got repos set-up
  + Source codes organized
  + Started merging development version of MF and GSFLOW into working version
  + Started merging PRMS5 into GSFLOW
* Got started on working on Sagehen 3 Layer model

**Accomplishments at 2nd code sprint (3/7/2017-3/10/2017, Aptos, CA):**

* *3/7/2017:*
  + Got started with repos on Monday
  + Merged various codes and made sure they compiled
  + Enrique and Eric started learning MMF routines
    - Started mimicking MMF/GSFLOW (C/Fortran) interfaces in simple examples
* *3/8/2017:*
  + Started working out mixed language communications with simple VB/C example projects in Vis Studio
  + Plan A: Attempted to wrap a C++ wrapper around the GSFLOW static library (\*.lib) file and the Fortran exe
  + 10pm quit time after failing to sort out Visual Studio’s use of underscores in naming convention of \*.dll and \*.lib files (Name of the project that we were experimenting with is located the GSFLOW\_examples repository and is called
  + More work on merging branches in repos (All)
  + Steve worked on compiling
    - PRMS5
    - Fortran version of MMF
    - Convert multiple parameter files into 1 (Sagehen example)
    - Got 128 HRU and “3 Layer” PRMS models going

* *3/9/2017*
  + Made this document
  + Worked on file that maps MODSIM link information to MODFLOW/PRMS isegs
  + Steve continued developing Fortran MMF
  + Group discussion on code architecture
    - Which terms should remain in GSFLOW control file and which should be made added to MODSIM custom code
    - Rich developed new GSFLOW functions for preparing aggregated flux for transfer between MODSIM and GSFLOW
  + Discussed the development of stream network mapping in PRMS. Needs to be something similar SFR approach
  + Discussed order of calls in various GSFLOW functions, and between MODSIM/GSFLOW