**Proposal for community development of RDBES package (“icesRDBES”)**

WKRDB-EST2, 15/9/2020

The aim of these guidelines has been not to make package development too onerous for the wider ICES community whilst also not putting too large a workload on the package maintainers – we also still need to ensure a minimum standard is met (e.g. the code is valid and in a reasonably consistent style, verifiably does what it’s supposed to, and can be submitted to CRAN).

Once the guidelines are agreed we should create a simple “How-to” document that can be given to potential contributors describing what they need to do.

* The package source code is hosted within its own repo by ICES e.g. in ices-tools-dev / ices-tools-prod
* A small group of maintainers will need to be volunteered, including someone from the Secretariat
* There are two branches within the repo: master and dev
* The master branch is protected so that only the maintainers can commit to it
* The dev branch is used for all development work – contributors can commit directly to it
* A “lint” tool is configured that will compare committed code to a defined style and warn if there any problems – contributors should endeavor to resolve any issues flagged
* A pull request needs to be created when we want to merge the development branch into the master branch – the maintainers will need to approve the pull request
* Release labels can be applied to the master branch to keep track of releases
* Contributors need to be given commit access to the repo – ICES have a work-flow for this.
* Contributors should pull the dev branch, make and commit changes on their local machine, and only push changes back to GitHub once their work is consistent (e.g. a new function is created and documented)
* The minimum standard of contribution is:
  + The contributor is using the latest version of R, roxygen and any dependencies
  + For each function an R file exists in the R directory
  + The Roxygen2 documentation comments have been generated for that function – the descriptions of functions and parameters should be written (what level of documentation do we require – can we have shorter descriptions but refer to a longer manual?)
  + Some simple examples of using the functions and its expected outputs are supplied (the maintainers will probably need to convert these into tests)
  + The code doesn’t have any major issues raised by the lint checks
* The “gold” standard of contribution is:
  + The contributor has defined tests for all new functionality
  + Devtools::check has been run successfully on their local machine
  + The code passes the automatic lint checks
* Periodically the maintainers will update the package in CRAN