

Extending CIG's Influence on Spreading Best Software Practices  
*Rene Gassmoller, University of Florida*

While CIG's technical infrastructure was originally designed to host and distribute geodynamic software, this role has largely been superseded by the availability of big online developer communities like Github and Gitlab, and archives like Figshare and Zenodo. It has become clear that CIG is no longer needed as a software archive, or host of software data. However, it remains essential to develop and implement FAIR standards for scientific software (findable, accessible, interoperable, reproducible), to ensure and improve software quality standards and to define and develop best practices in terms of software communities and software citation. In addition, the collection of CIG software has aged in the sense that many older projects outside of the flagship projects have become less active (a natural process in software lifecycles), while only few new packages have been added over the past 5 years.

In order to acknowledge this new environment, I would like to discuss removing the status of a "CIG community contributed code" and replace it with a more flexible and lightweight "CIG certified" software label. In order to increase the influx of new codes into the CIG community, contributors should not have to give up symbolic ownership of their code to CIG, instead CIG could develop a review, certification and support program that people can submit their codes to. Submitted codes would receive a formal review from CIG personnel or community members on how to improve their code and implement CIG's best practices after which (if properly implemented under guidance from CIG) they would receive the certification. This certification could be used to promote the software and to emphasize in funding proposals and journal publications that the software fulfills best practices. The certification could also act as a precondition for further CIG support (e.g. for developer meetings). All software (including CIG "developed", "certified", and "archived" inactive codes) should remain listed by CIG (to ensure findability), but their data should be transferred to proper data/software archives like Zenodo to keep CIG's technical infrastructure small and agile. CIG does not, and should not, "own" codes that have not been funded by CIG.

Computational methods and software development practices have undergone a paradigm shift since CIG was first created and the monolithic and planned development model that was part of CIG I has been completely replaced by rapid technological developments, like the agile development of small interacting units, cloud-based continuous integration and development services, the arrival of container technologies, web services, and the ever increasing role of the open-source movement. CIG IV can become the educational hub for further distributing these state-of-the-art methods among the geodynamics community, and it can be the support infrastructure that offers expertise in these areas to "CIG developed" and "CIG certified" software by providing resources and expertise on the path to better and more sustainable scientific software.