Contribution ID: **22** Type: **30 minute oral presentation**

Modernizing Scientific Software Development

Thursday, 27 September 2018 12:00 (30 minutes)

Modernizing Scientific Software Development

Christopher W. Harrop ,Cooperative Institute for Research in Environmental Sciences
Mark W. Govett, NOAA Earth Systems Research Laboratory

The commercial software industry has known for many years that the heavy cost of poorly designed software and undisciplined software development practices can sink an entire enterprise. To combat this problem, a mountain of software development tools and best practices have been developed and honed by the industry over the years. While no one tool or development methodology is objectively known to be a perfect solution, industry standard solutions have emerged that are widely adhered to.

The scientific software community, by contrast, has been slow to recognize and respond to the impact of poor software designs and development practices. There are several reasons for this, including insular attitudes and lack of software engineering expertise among scientists as well as a lack of incentives to create quality software in a research environment. As scientific software has increased in complexity, the problems stemming from poor designs and development practices have compounded. We are adapting techniques and technologies from the commercial software industry to rectify this situation.

We will present a description of software development design and processes that we have adapted and are adhering to in order to improve scientific software quality and the cost of its maintenance. We will describe how the Git-Flow branching model, used in tandem with an Agile-like development approach, affords us a structured, disciplined, means of ensuring integrity of software is maintained and that new developments are delivered in a timely fashion. We will describe a test-driven design approach, encompassing tests spanning fine and coarse granularities, scientific correctness, algorithmic correctness, and performance metrics, that ensures scientific code is tested before it is contributed and that it continues to be tested throughout the software lifecycle. Finally, we will show how we are adapting the use of design patterns and anti-patterns into our scientific software designs.

Affiliation

Cooperative Institute for Research in Environmental Sciences and NOAA Earth Systems Research Laboratory

Primary authors: HARROP, Christopher; Mr GOVETT, Mark (NOAA Earth Systems Laboratory)

Presenter: HARROP, Christopher

Track Classification: 18th Workshop on high performance computing in meteorology