Geophysical Research Abstracts
Vol. 18, EGU2016-9058-1, 2016
EGU General Assembly 2016

An integrated development workflow for community-driven FOSS-projects using continuous integration tools

Lars Bilke, Norihiro Watanabe, Dmitri Naumov, and Olaf Kolditz
Helmholtz Centre for Environmental Research - UFZ, Environmental Informatics, Leipzig, Germany

A complex software project in general with high standards regarding code quality requires automated tools to help developers in doing repetitive and tedious tasks such as compilation on different platforms and configurations, doing unit testing as well as end-to-end tests and generating distributable binaries and documentation. This is known as continuous integration (CI).

A community-driven FOSS-project within the Earth Sciences benefits even more from CI as time and resources regarding software development are often limited. Therefore testing developed code on more than the developers PC is a task which is often neglected and where CI can be the solution.

We developed an integrated workflow based on GitHub, Travis and Jenkins for the community project OpenGeoSys — a coupled multiphysics modeling and simulation package — allowing developers to concentrate on implementing new features in a tight feedback loop.
Every interested developer/user can create a pull request containing source code modifications on the online collaboration platform GitHub. The modifications are checked (compilation, compiler warnings, memory leaks, undefined behaviors, unit tests, end-to-end tests, analyzing differences in simulation run results between changes etc.) from the CI system which automatically responds to the pull request or by email on success or failure with detailed reports eventually requesting to improve the modifications. Core team developers review the modifications and merge them into the main development line once they satisfy agreed standards. We aim for efficient data structures and algorithms, self-explaining code, comprehensive documentation and high test code coverage.

This workflow keeps entry barriers to get involved into the project low and permits an agile development process concentrating on feature additions rather than software maintenance procedures.