Continuous Build, Test and Deployment - the Mantid Model

Mantid, our extensible framework for neutron and muon data reduction and analysis supports a community of X users across approximately Y instruments at two locations. At both facilities user beam-time is expensive and limited, so the software must work consistently, rapidly, and above all, provide the correct answers to a high degree of accuracy on every execution. Furthermore, our collaborating instrument scientists are often keen for us to quickly add new features to our existing code base, which is in excess of 1x106 lines of code. For this we need a team of T developers, situated at two facilities, 4000 miles and 7 hours apart. To compound issues, Mantid must build and deploy on a number of large range of completely different hardware and operating system environments. Continuous delivery of a project with these requirements is non-trivial.

Quality is best implemented from the ground up, so the Mantid project stipulates a combination of coding rules, and code-review to keep the code consistent and reduce errors and faults introduced by developers. In addition, code additions modifications are required to be unit-tested, and preferably made utilising Test Driven Development. Mantid uses a combination of Cxxtest and the Gmock Mocking framework to test the low-level functionality. Developers record, version and share their modifications using a distributed version control system called GIT, which allows them to work individually and also collaborate with the wider team.

With such a large number of incoming issues and a limited pool of resource to process the tasks, we utilise an issue tracking system called Trac to partition the work and prioritise it. We automatically cross-reference changes made in code base against individual tickets, to give a consistent story of the work completed.

A large portion of the work required in building, testing and deploying the project can be automated. Continuous integration and delivery is achieved using Jenkins. The Jenkins jobs are configured to not only build Mantid cross-platform, but also to run the entire unit test, performance test and python interface test suite every time a code change is made in Git.

For each instrument we support, we have at least one, “acceptance” workflow of operations that is representative of real usage on actual data. Every 24 hours, Jenkins will automatically clean-build, package, and deploy Mantid on every one of our target environments before running the entire test suite including these acceptance tests. Only once these pass is Mantid made available on our download page, and even then, it’s with a cautionary notice.

Official releases of Mantid occur every 2-4 months, and require a high degree of quality assurance. Every ticket recorded in Trac as completed for that release is individually tested and passed or failed. More recently, we have also been utilising a pool of instrument scientists to beta-test the candidate release.