ConTest: Introducing Continuous Integration and Test-Driven Development in a Software Engineering Course

As increasingly mission- and life-critical systems are developed (self-driving cars, Mars rovers), testing is rapidly growing in importance in industry. Software engineering techniques such as Test-Driven Development (TDD) and Continuous Integration (CI) are becoming more heavily used in industry as the demand for rapid development and code stability has increased. We believe it is therefore important for seniors enrolled in our university Software Engineering course (CS485) to get hands-on experience learning about and using these advanced testing techniques before graduating.

In our CS485 course we currently have large, ongoing in-house software project under development. In order for our students to learn and use TDD and CI in the classroom, we are currently developing an automatic-testing framework named ConTest that uses the Condor distributed execution system to run our regression-test suite on a heterogeneous computer cluster. The goal is that as student teams commit changes to a software versioning repository, the software system or a subsystem of it is immediately built if possible, and all of the tests in the regression test suite for that system are then run in parallel on a Condor cluster. The suite of pass/fail results is summarized and displayed on a web-based dashboard so that any regressions introduced by the commit are immediately apparent and can be quickly identified and fixed.

This paper describes the design and implementation of our ConTest CI testing framework and qualitatively evaluates our students' experiences in designing, implementing, and using the ConTest framework in our Software Engineering classroom. In particular, we evaluate the use of the system in developing our ongoing in-house software project, and how the rapid feedback provided by ConTest affects the students' software engineering practices.