Intelligent regression testing

POPULÄRVETENSKAPLIG SAMMANFATTNING **Edward Dunn Ekelund**

Intelligently selecting the most relevant tests by analyzing historical test data can cut down on the average time spent on regression testing by 95%, speeding up code integration by a factor of 20.

Automated regression testing is the practice of running tests that ensure nothing has been broken after changes have been made to the code, i.e. that no regressions have been introduced. In the 80s and 90s one of the most popular software project managing models was the *Waterfall* model. Waterfall is a sequential design process, meaning that all planning is done first, then the software is developed, and lastly it is verified by, for example, passing a regression test suite. In Waterfall, even if a test suite takes several hours to run it is not costly in relative terms since it is only run at the end of a project cycle which may have lasted several months or more.

Nowadays however, Waterfall has largely fallen out of favor, and so-called *Agile* processes have taken its place, where planning, development and verification are done iteratively and continuously. In other words, integration cycles are much shorter in Agile development and code changes are continuously verified several times per day. This means that a regression test suite that takes hours to complete has become massively inconvenient and costly.

My thesis work at Axis involved analyzing a vast database of historical regression test runs. Going through each test run in turn, I mapped the code that was changed to the tests that failed during the same run, which in the end resulted in a set of correlations between regression tests and code packages. Then, in future regression test runs, whenever code in a package is changed and has to be verified, we can use this correlation set to select only the regression tests that have historically proven to be affected by the changed package. All tests which are historically unaffected by changes in that particular part of the code are thus excluded from the regression test verification, and time is saved.

Evaluation of my results have shown that when running only the intelligently selected tests we find nearly as many regressions as when running the full regression test suite. Yet, since the test set is much smaller, time spent on regression test verification is only 5% of the time compared to running the full test suite. In Axis' case it means going from an 8 hour regression test suite to one that on average takes less than 20 minutes to complete.

The consequence of cutting down on time wasted on running unnecessary tests is that developers no longer need to wait a full work day to verify code changes. The rate of integration can be sped up by a factor of 20. With more rapid code integration, it takes less time to fix bugs and introduce new features, which leads to more highly valued software, and, ultimately, increased revenue for the company.