

Broadly, agile software development is an approach where code is frequently built, tested and shipped, leading to short release cycles. Extreme version is the DevOps approach where the development, testing and deployment pipelines are merged and software is continuously tested and updated.

In this context our work focuses on identifying spots where the participants should be more aware of the performance and offers approaches and tools to improve their awareness with the ultimate goal of producing better software in shorter time. In general, the awareness is raised by testing, documenting, and monitoring the performance in all phases of the development cycle.

In this thesis we (1) show a framework for writing performance tests for individual components (e.g. libraries). The tests capture and codify assumptions about the performance into runnable artifacts that simplify repeatability and automation. For evaluation of the performance tests we (2) propose new methods, which can automatically detect performance regressions. These methods are designed with inherent variation of performance data in mind and are able to filter it out in order to detect true regressions. Then we (3) reuse the performance tests to provide the developers with accurate and up-to-date performance API documentation that steer them towards writing efficient solutions from the beginning. Finally we (4) explore the practical limits of using runtime monitoring to gather performance data from production environments. This allows us to assess whether it is possible to use production performance data in the performance tests and thus validate that the assumptions also hold outside – inherently simplified – testing environment.