prof. RNDr. Jan Kratochvíl, CSc.
Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic

Priv.-Doz. Mag. Dr.
Rick Rabiser
Senior Scientist
Christian Doppler Lab MEVSS
Institute for
Software Systems Engineering

P +43 732 2468 4363
F +43 732 2468 4345
rick.rabiser@jku.at

Office:
Birgit Kranzl
P +43 732 2468 4341
F +43 732 2468 4345
birgit.kranzl@jku.at

Linz, April 18, 2018

RE: Review of Thesis "Performance Awareness in Agile Software Development" by Vojtěch Horký

This thesis presents work conducted in the area of performance testing and monitoring to support (agile) software development. The ultimate goal of all performance testing and monitoring is to ensure and improve the quality of software and avoid failures with a wide variety of possible negative effects ranging from annoyed customers to financial losses to even more serious problems, e.g., caused by crashing software. In agile software development, especially due to the recent trend towards continuous delivery and DevOps, software is frequently built, tested and deployed, making the quick identification of performance issues even more critical than in "traditional" software development.

Existing approaches largely focus on functional testing and those that do focus on performance testing mostly aim at system-wide or at least more large-scale performance testing. The approach proposed in this thesis aims at enabling performance testing of individual components, especially automating performance testing by generating performance tests for individual components and automatically measuring performance regressions. In short: the proposed approach aims at quickly and automatically detecting performance issues on the level of individual components, similar to unit tests in functional testing. In addition, the thesis also aims at generating performance documentation for software developers to help them during maintenance as well as when writing new software. Again, state-of-the-art code documentation approaches mostly only focus on functional documentation. The work presented in this thesis thus aims to provide additional useful performance-related information to developers of individual components. In addition, the thesis also explores the limits of runtime monitoring to provide performance tests with production data.

The thesis is a cumulative thesis (i.e., comprises and summarizes publications co-authored by the thesis author) structured in three parts and nine chapters (not counting tables of contents and the bibliography):

- Part I – Introduction and Contribution Overview – comprises three chapters that provide a detailed overview of the contributions of the thesis, i.e., cross-cutting the three publications presented in Part II
    - Chapter 1 – Performance Awareness in Software Development – introduces the concept of performance awareness and motivates the overall work. It also explains key areas such as (performance/code) documentation, build-time testing, automated regression detection and runtime software monitoring.
    - Chapter 2 – Goals in Detail – lists the four main goals of the thesis: G1: come up with an approach for writing performance tests for individual software components; G2: develop a method for the robust detection of performance regressions; G3: develop an approach to generate performance documentation for individual software components; and G4: explore the limits of runtime performance monitoring.
    - Chapter 3 – provides a detailed overview of the four contributions of the thesis that directly map with the four stated goals, i.e., Performance Unit Testing; Robust Regression Detection; Performance API Documentation; Performance Monitoring Overhead. The chapter explains for each contribution what was developed and how the developments were evaluated.
- Part II – Collection of Papers – starts with a brief overview and then contains (in three chapters) the three publications that constitute the cumulative thesis, i.e., one Journal Article published in 2017 in the Journal on Automated Software Engineering and two papers published at the International Conference on Performance Engineering.
- Part III provides a discussion of related work (chapter 8) and summarizes the thesis (chapter 9). It clearly re-iterates the key contributions and also discusses future work that could be conducted based on the results of the thesis.

In the reviewer's opinion, this thesis provides two core contributions (new scientific results): (1) a framework to specify performance unit tests using stochastic performance logic (SPL) and to automatically detect performance regressions using workload generators (existing benchmarks can also be used as workload generators) and (2) a prototype to automatically generate performance documentation based on this framework. The prototype can support the developer in maintaining code as well as developing new code, as it shows him/her immediately what effects certain code can have (has) on the performance of the software. Both contributions were prototypically implemented, by the thesis author and his students, in the Java realm, i.e., SPL can be added to Java methods using annotations, workload generators can be written in Java, and performance documentation is generated as/using JavaDoc annotations. The wide use of Java makes this choice reasonable, even though it would have been interesting to discuss the potential for implementing the framework also in other technologies and on other platforms.

In addition to these contributions, the thesis also discusses the limits of runtime monitoring and makes clear the limiting factors, also for the presented contributions, i.e., precision of software sensors (probes) and complex interactions across the full hardware and software stack. In this direction, it would have been interesting to see more experiments. The presented experiments provide first insights, but this is something, as also stated in the conclusions, that needs to be continued in future work.

While the presented framework and prototypical implementations seem very reasonable and useful, and while the presented experiments demonstrate its feasibility, one key hypothesis already part of the thesis title – i.e., improving/increasing performance awareness of developers – could not be fully validated. While a study has been conducted with students, the data did not allow drawing statistically significant conclusions due to a high variance of results. It is very good and honest that this limitation is explicitly pointed out in the thesis and that it was not tried to hide this fact. While it is clear that more experiments and user studies are required to prove the proposed approach is useful in practice, I think its basic feasibility and potential usefulness could indeed be demonstrated in the thesis, respectively in the three publications, in a satisfying way.

Overall, the thesis is concerned with a current, interesting and relevant research topic of high practical relevancy for engineering software systems using state-of-the-art development approaches such as agile development. The work provides new and essential contributions in this direction as summarized above.

Also, the three publications the thesis comprises have been published in venues that are both, highly relevant for the research area as well as of high quality, i.e., the Journal on Automated Software Engineering is a top international journal in the general area of software engineering and the International Conference on Performance Engineering is the top venue in the area of performance engineering. Unfortunately, it remains unclear which author contributed what to which publications, which is complicated by the fact that the authors seems to have been sorted alphabetically. The track record of the thesis author – e.g., on DBLP and Google Scholar, however, shows that the thesis author already published 14 publications (one journal article, 11 conference papers, 2 book chapters), which is quite impressive for a PhD student. Many of these publications are very closely related to the PhD thesis and the thesis author also already has been cited several times. This demonstrates that the thesis author can work and publish on an international level and that his work had an impact that goes beyond the performance engineering community.

The prototypes further demonstrate that the thesis author was capable of providing very complex implementations using state-of-the-art technologies. Conducting the experiments presented in the thesis required both knowledge about state-of-the-art technologies as well as research methods (experiments, benchmarks, statistical methods, user studies). All this shows that the thesis author can address scientific problems independently in an important research field of high practical relevancy.

Overall, the thesis proves the author's ability for creative scientific work very well. I thus recommend the acceptance of this doctoral thesis and rate it with an A (1) grade (best possible grade).

Best regards,

Rick Rabiser